electronic bookmarks

alimentatori radiofrequenza Robotica microcontrollore in VHDL Decoder 7 segmenti trasmettitore DSB circuiti integrati alimentatori Digitale terrestre

n. 294/295 DICEMBRE/GENNAIO Anno XXV
solo Fare Elettronica - € 6,00
Fare Elettronica + Mr A. Keer € 7,90 in più

fe

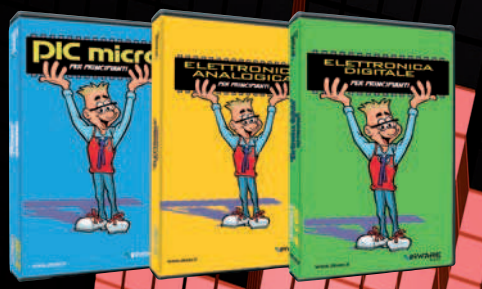
fare elettronica
www.farelettronica.com

PROGETTI

- Trasmettitore DSB per i 20m
- Cinque melodie in un PIC
- Smartcard reader
- Robot cingolato

PER IMPARARE

- *Il protocollo SIP nei sistemi VoIP*
- *Decoder 7 segmenti con memoria*
- *I resistori elettronici*
- *Lezioni di VHDL*



Richiedi al tuo edicolante i tre DVD di Mr A.Keer!

Il digitale terrestre

la tecnologia e i decoder

FREQUENZIMETRO DA 60 MHZ

con PIC16F628

IE mobile

Inquadra il codice per accedere ai contenuti!

ISSN 1591-2272
00295
9 771591 227008

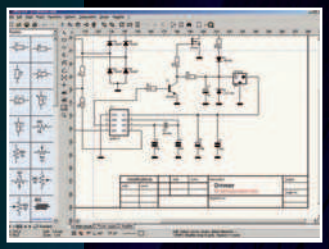
MENSILE Poste Italiane Spa - Spedizione in abbonamento Postale - D.L. 353/2003 (conv.in L.27/02/2004 n.46) art.1, comma1 DCB Milano. In caso di mancato recapito, restituire all'editore che si impegna a pagare la relativa tassa presso il CPM di Rozzano - Milano

A VOLTE RITORNANO...

I migliori software per l'elettronica

sPlan 6.0

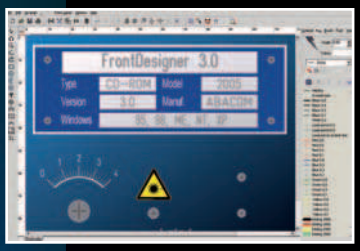
Programma per disegnare gli schemi elettrici.



Solo € 47,88

Front Designer

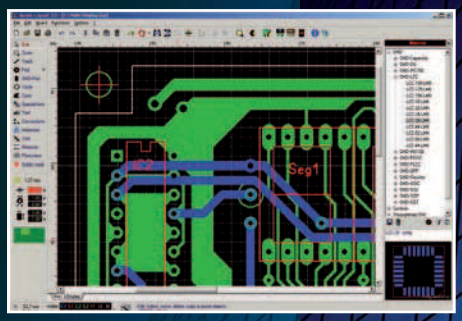
Programma per disegnare i pannelli frontali



Solo € 47,88

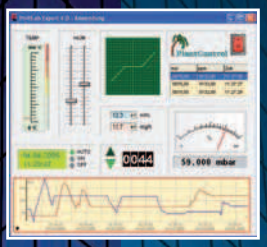
Sprint-Layout 5.0

Con Sprint-Layout è possibile disegnare il proprio PCB in modo facile e veloce.



Solo € 47,88

ProfiLab-Expert 4.0



Il software ProfiLab-Expert permette di sviluppare i propri progetti di misurazione digitale o analogica.

Solo € 119,88

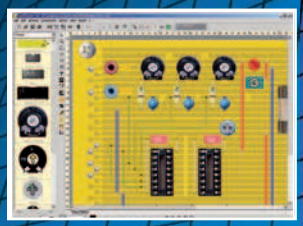
AudioWave 2.0

Con AudioWave 2.0 la propria scheda audio diventa un confortevole generatore di segnale LF che genera segnali da 1Hz a 20 kHz con una risoluzione di 1Hz.

Solo € 35,88

LochMaster 3.0

LochMaster 3.0 è uno strumento di sviluppo per la progettazione di boards. Possiede utili funzionalità per disegnare, documentare e testare una board.



Solo € 47,88

Prezzi IVA inclusa

PUOI ORDINARE I SOFTWARE ABACOM SUL SITO
www.ieshop.it/abacom
oppure telefonando allo 02 66504755

CODICE MIP 2779018

IN EDICOLA

200 progetti per 2000 applicazioni

Il nuovissimo numero di

Fare Elettronica EXTRA

INWARE EDIZIONI

CODICE MIP 2779026

da non perdere!

sommario

294/295
dicembre/gennaio 2009/2010

Zoom in

18 IL DIGITALE TERRESTRE
Un cambiamento radicale di uno degli oggetti più utilizzati e maggiormente diffusi nel mondo. In questo articolo cercheremo di scoprire gli aspetti tecnici della televisione digitale, partendo dalla codifica dei segnali audio/video, passando per la modulazione, la trasmissione radio terrestre e la ricezione su un set-top-box.
di Giuseppe De Gennaro

Progettare & costruire

30 CINQUE MELODIE CON UN PIC
Questo mese realizziamo un progetto sonoro nel quale utilizziamo un microcontrollore PIC abbinato al nuovo compilatore MikroBasic Pro della MikroElektronika per riprodurre ben cinque melodie musicali
di Giovanni Di Maria

36 FREQUENZIMETRO CON PIC 16F628
Strumento "minimo" con un'ottima versatilità, è utilizzabile sia come strumento a sé stante, sia come sintonia elettronica per piccoli ricetrasmettitori auto costruiti.
di Daniele Cappa

42 LETTORE DI SMARTCARD (parte II)
In questa seconda e ultima parte riprendiamo il progetto del lettore standalone di smartcard dotato di display e tastierino per l'utilizzo in applicazioni di fidelizzazione della clientela. Dopo aver realizzato il prototipo completo, andremo qui a discuterne il funzionamento e il firmware che lo anima
di Giuseppe Modugno

Primi passi

54 I RESISTORI ELETTRONICI
I resistori, chiamati spesso e impropriamente resistenze, sono componenti fondamentali nella realizzazione di qualunque circuito elettronico. Sono prodotti in un'ampia varietà di forme, di valori ohmici e di potenze. Vediamo quali sono gli aspetti tecnologici che li caratterizzano e li diversificano e quali sono le codifiche impiegate per identificarli tecnicamente
di Antonio Giannico

L'angolo di Mr A.Keer

66 DECODER "CON MEMORIA"
Con i decoder "con memoria" si conclude la rassegna dei principali dispositivi in grado di controllare i visualizzatori formati da digit "a 7 segmenti". Scopriamone i segreti e il modo migliore per utilizzarli
di Giorgio Ober

Imparare & approfondire

74 INTRODUZIONE AL VOIP (parte iV)
In questa quarta e ultima puntata affronteremo insieme il protocollo SIP. Esso rappresenta la tendenza più attuale nella progettazione di sistemi VoIP
di Grazia Ancona

86 Lezioni di VHDL (parte VIII) SILICORE SU SPARTAN
Nella scorsa puntata ci siamo soffermati sul processore SLC1657, un RISC a 8 bit. Il processore, realizzato interamente in VHDL e diffuso da Silicore con la solita licenza GPL, risulta particolarmente indicato per applicazioni di controllo.
di Francesco Pentella

Rispondi e . . .**VINCI!** *pag.* **90**

DIVERTITI E METTI ALLA PROVA LE TUE CONOSCENZE CON **ELETTRO QUIZ** E VINCI OGNI MESE ESCLUSIVI PREMI!

Radio & radio

92 TRASMETTITORE DSB-SC PER I 20 METRI

Se siete in procinto di prendere la patente di radioamatore e volete fare un po' di pratica di laboratorio prima di affrontare gli esami, ecco un piccolo progetto che vi dà la possibilità di testare su campo quanto avete studiato sui libri.

di Remo Riglioni (IZ0OPG)

Robot Zone

98 ROBOT KIT(parte I)

Progettare piccoli robot domestici è certamente un'attività gratificante che può riempire di gioia qualsiasi appassionato. Come ogni attività umana, e in modo particolare un lavoro di progettazione integrata hardware e software, è necessario definire, in modo chiaro e preciso, le diverse fasi di lavoro: solo in questo modo possiamo raggiungere il nostro obiettivo

di Francesco Pentella

rubriche

7 Editoriale
10 Idee di progetto
14 Eventi
16 News
90 Elettroquiz
91 Luditronica
104 Thread Focus

LUPUS IN FABULA

Fairchild 16 - Istituto Italiano di Tecnologia di Genova 16 - LEM 16 -Linear Technology 16 - Microchip 30, 36 -Università di Pisa 16

COMING...

...prossimamente su Fare Elettronica

- **PROGETTARE CON RFID**
- **SENSORE DI FUMO CON MSP430**
- **TUTTO SULLE LAMPADE A FLUORESCENZA**
- **TRIAC ED SCR, COME FUNZIONANO?**

facebook

IE mobile

Fare Elettronica è su Facebook!!

Unisciti al gruppo

GLI ARTICOLI CONTRASSEGNATI COL SIMBOLO SONO GIÀ DISPONIBILI IN FORMATO PDF* ALL'INDIRIZZO WWW.FARELETTRONICA.COM/CLUB

*Puoi iscriverti al CLUB di Fare Elettronica versando una piccola quota annuale.

IE mobile

Usa il mobile tagging per vedere sul tuo telefonino gli approfondimenti e gli aggiornamenti sui contenuti della rivista. Per saperne di più visita *www.inwaredizioni.it/IEMOB*

COSA DEVI FARE

1) Questa operazione la farai una sola volta. Scarica ed installa gratuitamente il software per il tuo telefonino dalla pagina http://gettag.mobi
2) Clicca l'icona TagReader sul tuo telefonino
3) Inquadra il codice e segui le istruzioni del software di lettura
4) Il telefono si collega al sito mobile per visualizzare i contenuti

elenco inserzionisti

Blu Nautilus pag. 9, 53, 85
Piazza Tre Martiri 24 - 47900 Rimini (RN)
Tel. 0541 439575 - www.blunautilus.it

Comune di Scandiano - Ufficio Fiera pag. 15
Piazza Trampolini 1 - 42019 Scandiano (RE)
Tel. 0522-764290 - www.fierascandiano.it

E.R.F. pag. 61
Largo Fiera della Pesca 11 - 60100 Ancona (AN)
Tel. 0733 780815 - www.erf.it

Exposition service. pag. 14
V.le Dante Alighieri, 54 – 48022 Lugo (RA)
Tel. 0545 27548 – www.mondoelettronica.it

Farnell Italia pag. 6, 23
C.so Europa, 20-22 – 20020 Lainate (MI)
Tel. 02 939951 - www.farnell.com

Futura Elettronica pag. 45, 79
Via Adige, 11 - 21013 Gallarate (VA)
Tel. 0331 792287 - www.futuranet.it

Inware pag. IV cop
via Cadorna, 27/31 - 20032 Cormano (MI)
Tel. 02 66504794 - www.elettroshop.com

LeCroy pag. 41
via E. Mattei Valecenter 1/c/102a - 30020 Marcon (VE)
Tel. 041 5997011 - www.lecroy.com

Micromed pag.63
Via Valpadana 162B/2 - 00141 Roma
Tel. 06 82000066 - www.micromed.it

MikroElektronica pag. 35
Visegradska, 1A - 11000 Belgrade
Tel. +381 11 3628830 – www.mikroe.com

Millennium Dataware pag. 89
Corso Repubblica 48 - 15057 Tortona (AL)
Tel. 0131 860254 - www.mdsrl.it

Nital pag. 96
Via Tabacchi 33 - 10132 Torino (TO)
Tel. 011 8144332 - www.irobot.it

PCB Pool pag. 17
Bay 98-99 Shannon Free Zone
Shannon - County Clare
Tel. 02 64672645 - www.pcb-pool.com

Tecnoimprese pag. 57
V. Console Flaminio, 19 - 20134 Milano (MI)
Tel. 02 210111244 - www.tecnoimprese.it

Teltools pag. 16
Via della martinella 9 - 20152 Milano
www.carrideo.it - www.teltools.it

element14

www.element-14.com

Il portale tecnico / la comunità elettronica per i progettisti

La comunità elettronica si incontra su element14

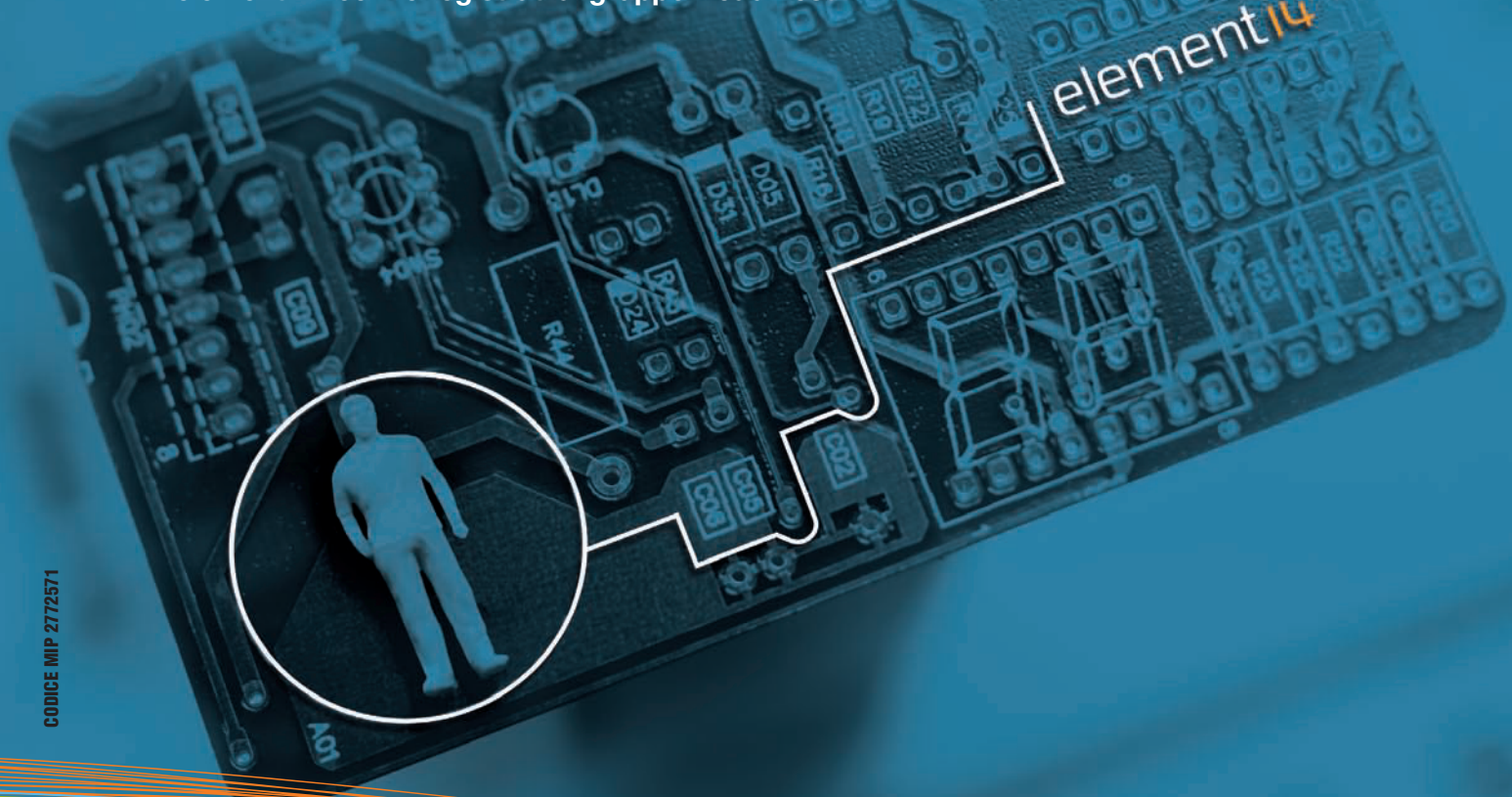
Una risorsa indipendente e 'all-in-one' per la ricerca di informazioni e strumenti per la progettazione e per la collaborazione fra i progettisti elettronici.

Vinci Kit di sviluppo e altri componenti elettronici!

Partecipa al ROAD TEST e avrai la possibilità di ricevere in omaggio kit di sviluppo e componenti. Fino al 30 gennaio 2010 potrai ricevere:

- RFMD RF205x Development Kit
- Atmel Xplain Evaluation and Demonstration kit

Partecipare è semplice: per scoprire i dettagli visita www.element-14.com e registrati al gruppo Road Test.



CODICE MIP 2772571

element14

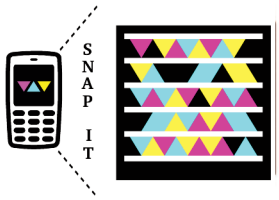
editoriale

di MAURIZIO DEL CORSO

Frequenze LIBERE!

Con l'avvento al digitale terrestre e il progressivo switch-off dei canali analogici si libereranno molte frequenze, ma come verranno impiegate?

In gergo tecnico è noto come "dividendo digitale" ed è la disponibilità di frequenza fino ad oggi occupate dalla TV analogica e rese libere con il passaggio al digitale terrestre. Lo spettro radio liberato consente la trasmissione di dati anche su lunghe distanze ed è quindi una risorsa molto preziosa per la quale la Commissione Europea sta determinando come sfruttarla al massimo. I potenziali usi del dividendo digitale sono molteplici e focalizzati sui servizi di consumo di massa come quelli mobili in banda larga. È importante però che vi sia una azione coordinata tra tutti i paesi europei in quanto se una stessa apparecchiatura può operare in tutti i paesi europei si genera un enorme vantaggio economico derivante da economie di scala per i produttori, con conseguente abbassamento dei prezzi, aumento della domanda e maggiore accessibilità ai nuovi servizi da parte dei consumatori. La destinazione delle nuove frequenze è ancora oggetto di discussione per cui non ci resta che attendere, nel frattempo possiamo approfondire le nostre conoscenze sulla tecnologia digitale terrestre con l'articolo di pagina 18. Per il vostro laboratorio abbiamo preparato il progetto di un frequenzimetro da 60MHz che vi sarà senz'altro utile, mentre per gli amanti della robotica, troverete in RobotZone il progetto di un robot che potrete dotare di cingoli ed altri add-on. Questo mese abbiamo anche riservato spazio a chi è alle prime armi con una trattazione esaustiva sui resistori.



Scarica l'applicazione per il tuo cellulare su http://gettag.mobi

Scatta una foto al codice sopra riportato i primi 10 avranno una iscrizione gratuita al club di Fare Elettronica!

in fatto di "campagne"... la nostra non ha rivali

ABBONATI
SUBITO
3 NUMERI
SONO GRATIS !!!

DIRETTORE RESPONSABILE
Antonio Cirella

DIRETTORE TECNICO
Maurizio Del Corso

Segreteria di redazione
Giorgia Generali

Coordinamento editoriale
Rachele Villa

Comitato Scientifico
Simone Masoni (Microtest), Francesco Picchi (Microtest), Massimo Rovini (Università degli Studi di Pisa), Tiziano Galizia (Tigal), Claudio Turchetti (Università Politecnica delle Marche).

Art Director
Patrizia Villa

Grafica e impaginazione
Mara Florian

Hanno collaborato in questo numero:
Grazia Ancona, Daniele Cappa, Giuseppe De Gennaro, Giovanni Di Maria, Antonio Giannico, Giuseppe Modugno, Giorgio Ober, Francesco Pentella, Remo Riglioni.

Direzione Redazione Pubblicitá International Advertisement
INWARE Edizioni srl
Via Cadorna, 27/31
20032 Cormano (MI)
Tel. 02.66504755
Fax 02.66508225
info@inwaredizioni.it
www.inwaredizioni.it
Redazione: fe@inwaredizioni.it

Stampa
Grafiche Sima
Via Provinciale Francesca, 52
24040 Ciserano (BG)

Distribuzione
Parrini & C s.p.a.
Via di Santa Cornelia, 9
00060 Formello (RM)

Ufficio Abbonamenti
INWARE Edizioni srl
Via Cadorna, 27/31
20032 Cormano (MI)
Per informazioni, sottoscrizione o rinnovo dell'abbonamento:
abbonamenti@inwaredizioni.it
Tel. 02.66504755
Fax. 02.66508225
L'ufficio abbonamenti è disponibile telefonicamente dal lunedì al venerdì dalle 14,30 alle 17,30.
Tel. 02.66504755
Fax 02.66508225

Abbonamento per l'Italia:
€ 49,50
Abbonamento per l'estero:
€ 115,00
Gli arretrati potranno essere richiesti, per iscritto, a **€ 9,00** oltre le spese di spedizione

Autorizzazione alla pubblicazione
Tribunale di Milano n.647
del 17/11/2003

© Copyright
Tutti i diritti di riproduzione o di traduzione degli articoli pubblicati sono riservati. Manoscritti, disegni e fotografie sono di proprietà di Inware Edizioni srl. È vietata la riproduzione anche parziale degli articoli salvo espressa autorizzazione scritta dell'editore. I contenuti pubblicitari sono riportati senza responsabilità, a puro titolo informativo.

Privacy
Nel caso la rivista sia pervenuta in abbonamento o in omaggio, si rende noto che i dati in nostro possesso sono impiegati nel pieno rispetto del D.Lgs. 196/2003. I dati trasmessi a mezzo cartoline o questionari presenti nella rivista, potranno venire utilizzati per indagini di mercato, proposte commerciali, o l'inoltro di altri prodotti editoriali a scopo di saggio. L'interessato potrà avvalersi dei diritti previsti dalla succitata legge. In conformità a quanto disposto dal Codice di deontologia relativo al Trattamento di dati personali art. 2, comma 2, si comunica che presso la nostra sede di Cormano Via Cadorna 27, esiste una banca dati di uso redazionale. Gli interessati potranno esercitare i diritti previsti dal D.Lgs. 196/2003 contattando il Responsabile del Trattamento Inware Edizioni Srl (info@inwaredizioni.it).

Collaborare con FARE ELETTRONICA
Le richieste di collaborazione vanno indirizzate all'attenzione di Maurizio Del Corso (m.delcorso@inwaredizioni.it) e accompagnate, se possibile, da una breve descrizione delle vostre competenze tecniche e/o editoriali, oltre che da un elenco degli argomenti e/o progetti che desiderate proporre.

Expo Elettronica CREMONA

12-13 dicembre 09

Cremona Fiere

ore 9 / 18

AFFARI LAST MINUTE

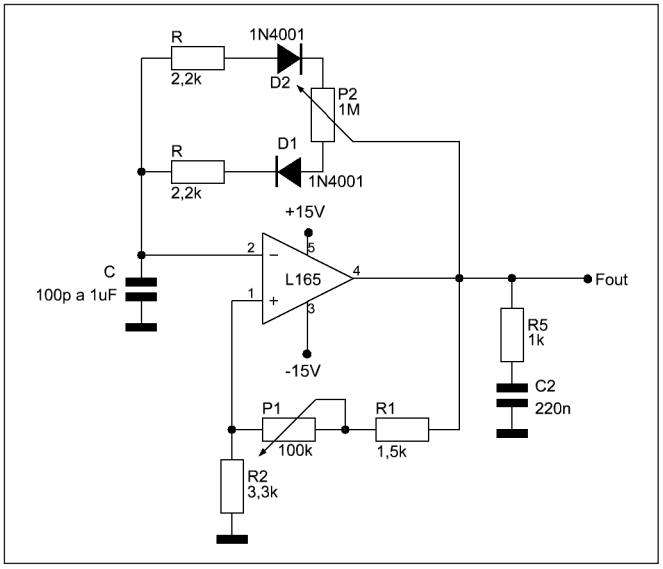
MOSTRA MERCATO di informatica, telefonia mobile fissa, hardware e software, ricezione satellitare, hobbistica, DVD e Games, materiali di consumo, elettronica

MERCATINO DELL ' USATO

VALE COME RIDOTTO

Organizzazione e informazioni: BLU Nautilus tel 0541 439573 info@expoelettronica.it • www.expoelettronica.it

rif. Fare Elettronica CODICE MIP 2754293



ASTABILE REGOLABILE

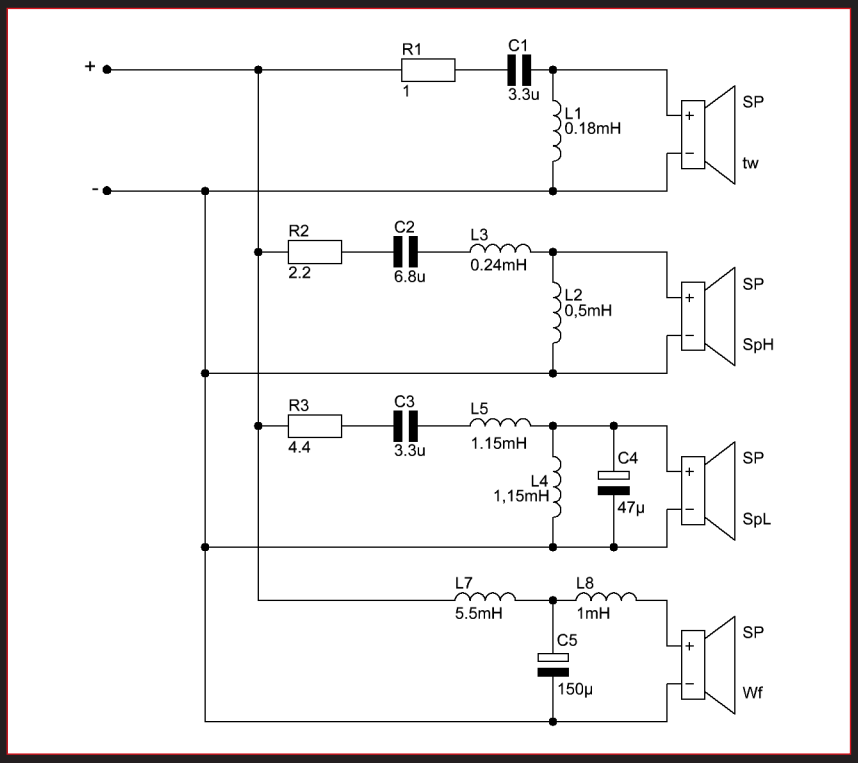
Il circuito è il classico oscillatore astabile provvisto di regolazione indipendente del duty cycle "P2" e della frequenza "P1". Per ottenere una frequenza di oscillazione di 700 Hz dobbiamo collegare in C1 un condensatore da 10 nF e il potenziometro P1 deve fornire una resistenza di 100 kohm, mentre per una frequenza di 25 Hz dobbiamo collegare un condensatore da 10 nF e il potenziometro P1 deve essere regolato a 0 ohm. Per ottenere altri valori di frequenza basta seguire le seguenti formule:

$$f = 2RC \ln \frac{1+\beta}{1-\beta}$$

e per definire il beta, ovvero il guadagno, utilizziamo la seguente formula:

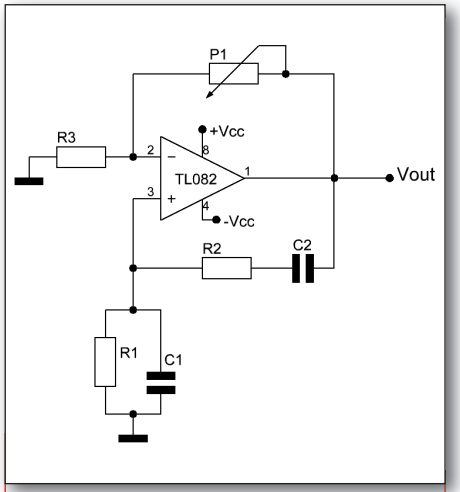
$$\beta = \frac{R2}{R1+R2}$$

Il circuito deve essere alimentato con una tensione duale di 15 V.



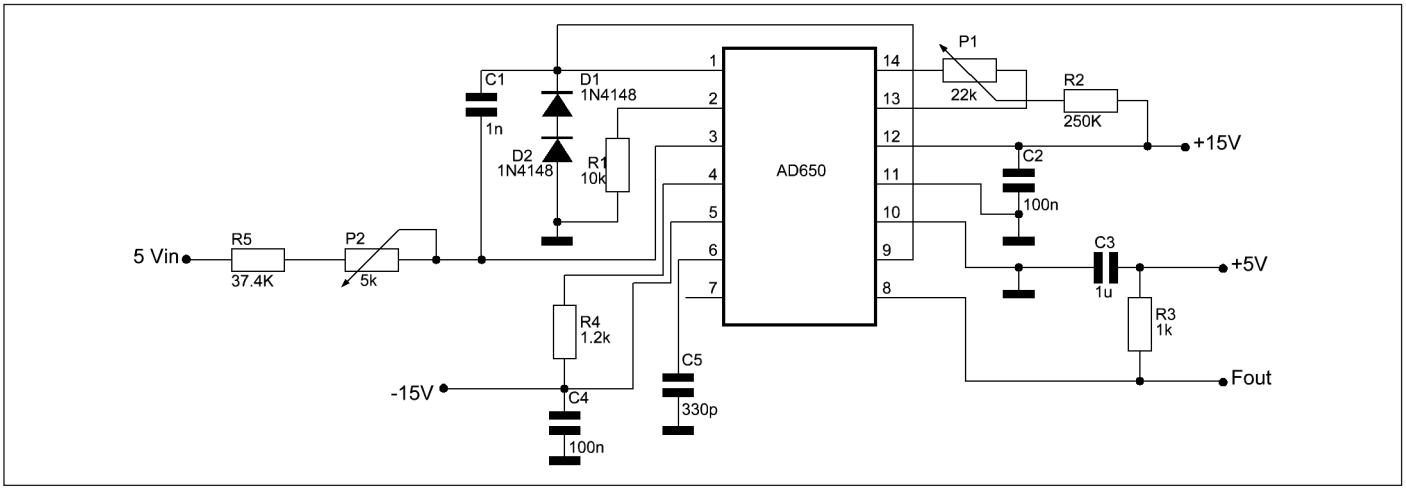
CROSSOVER A 4 INGRESSI

Questo circuito è formato da 4 filtri del secondo, terzo e quarto ordine. Tale circuito viene impiegato nei comunissimi altoparlanti a colonna degli impianti stereo. Le frequenze di taglio dei filtri sono: 350 Hz, 1500 Hz e 6000 Hz. A ogni uscita del filtro corrisponde un particolare tipo di cassa: per primo abbiamo il Tweeter (Tw), come secondo e terzo abbiamo comuni casse per i medi (Mh e Mb) e infine subwoofer (Wf).



OSCILLATORE A PONTE WIEN

Nello schema è riportato un oscillatore a ponte di Wien. Esso è costituito da un amplificatore operazionale TL082, da condensatori e induttanze.
Il segnale in uscita è di tipo sinusoidale e dipende dai parametri C ed R. Per il calcolo della frequenza si usa la formula f = 1/2πRC, mentre a provvedere all'ampiezza del segnale abbiamo la formula R2 = 2R1.

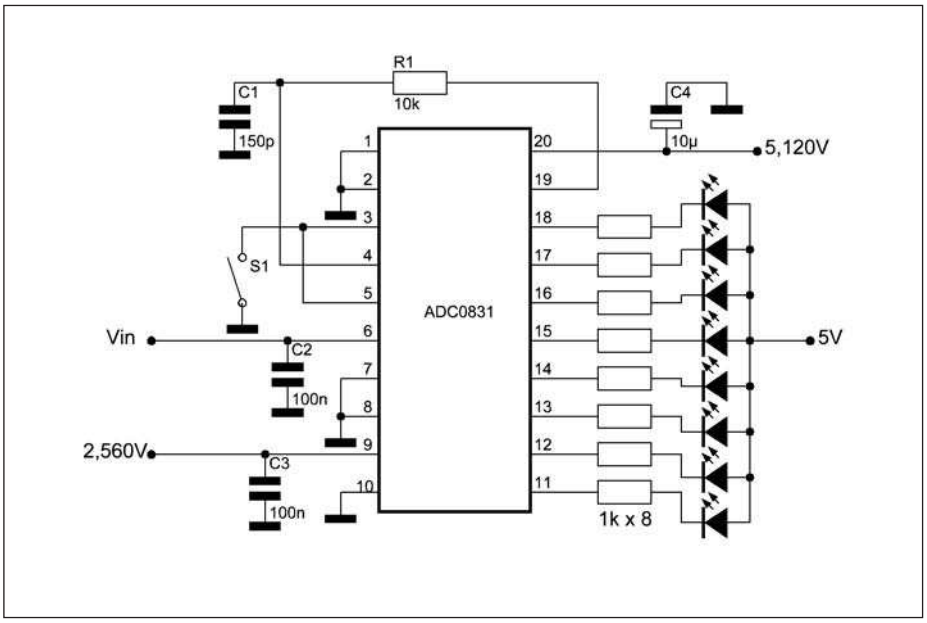


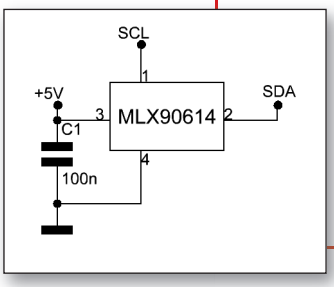
CONVERTITORE *TENSIONE/FREQUENZA*

L'AD650 prodotto dalla Analog Device è un convertitore V/F e F/V; esso sfrutta la proprietà del bilanciamento di carica. Uno degli schemi applicativi è quello riportato in figura che offre la possibilità di ottenere 100 kHz con una tensione di ingresso di 5 V. Con tale dispositivo si possono tuttavia ottenere molte configurazioni e ottenere tensioni di uscita e frequenze di uscita a piacimento. A questo proposito il manuale offre formule e diagrammi appositi. Il circuito deve essere alimentato con una tensione duale di 15 V.

Conversione AD

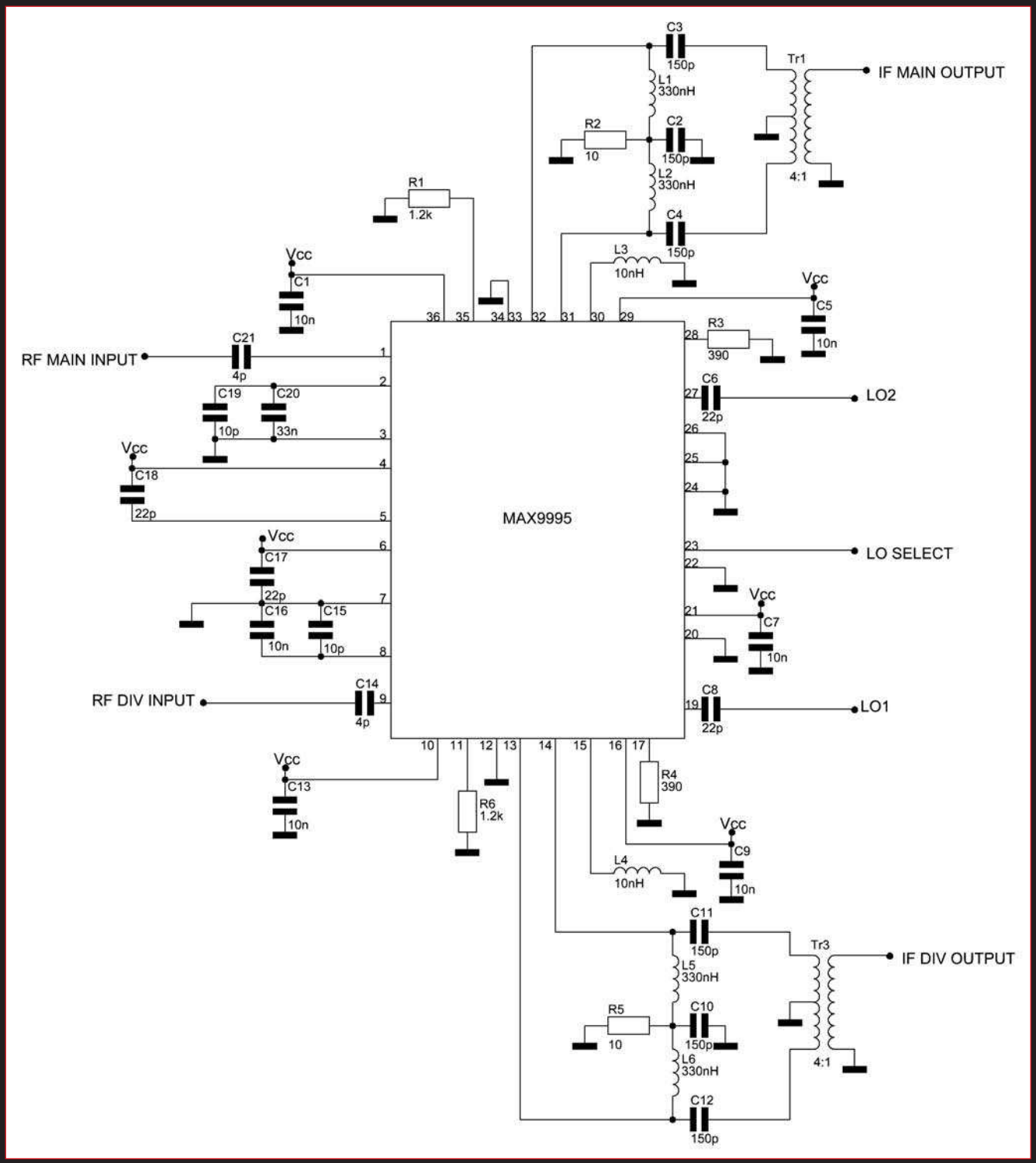
L'ADC0801 è un convertitore A/D piuttosto comune prodotto dalla National Semiconductor del tipo ad approssimazione successiva. Realizzati in tecnologia CMOS, essi presentano una risoluzione a 8 bit con un tempo di conversione a 100 us. I terminali di uscita sono di tipo tri-state con livelli TTL compatibili. Il segnale di clock può essere fornito dall'esterno o può essere generato dall'integrato stesso come in figura. Questo schema permette di capire come avviene la conversione facendo arrivare sul pin 6 una tensione regolabile da 0 a 5 V; facendo così il codice binario viene indicato dai LED che si accendono e si spengono.





TERMOMETRO *A INFRAROSSI*

L'MLX90614 è un termometro a infrarossi senza contatto per misurazioni di temperatura. Questo sensore è costituito da un sensore IrDa sensibile alla temperatura. Il dispositivo viene calibrato in fabbrica secondo un sistema PWM per essere sensibile a temperature da -20°C a 120°C. Internamente è costituito da un convertitore da 17 bit che lo rende adatto per misure di precisione. Infatti viene usato per misurare le temperature ai bambini con età inferiore ai 2 anni e anche in applicazioni militari.



CONVERSIONE AD ALTA LINEARITÀ

Il MAX9995 è un convertitore ad alta linearità prodotto dalla Maxim. Fornisce un guadagno 6.1 dB, 25,6 dBm IIP3 e 9.8 dB. Questi dispositivi sono disponibili in un compatto 36-pin sottile. La tensione di alimentazione si aggira intorno ai 5 V e dissipa una corrente di 350 mA. In figura è riportata la connessione tipica raccomandata dalla casa produttrice che garantisce un buon funzionamento. Questo integrato viene utilizzato per applicazioni militari, nei sistemi radio, wireless, ecc.

CARPE DIEM!

NON PERDERE IL NUMERO DI QUESTO MESE

CODICE MIP 2779038

NUOVA ORGANIZZAZIONE

FIERA DELL'ELETTRONICA
INFORMATICA E RADIANTISMO

Mostra Mercato Mondo Elettronica.net

Informatica - PC - Hardware
Software - Telefonia - Digitale
Surplus - TV-SAT
Videogames - Componentistica
LCD - Hobbistica - Accessori
Materiali di consumo
Radiantismo - Dischi e CD
HI-FI - Mp3 - Mp4

13-14 FEBBRAIO

FIERA DI FERRARA
Via della Fiera, 11
Autostrada A13 uscita "Ferrara sud"

ORARIO 9-18,30

NUOVA ORGANIZZAZIONE

exposition SERVICE
Cell. 348 9030832
info@expositionservice.it

www.mondoelettronica.net

VALE COME RIDOTTO
ANCHE IN FOTOCOPIA

12-13 dicembre 2009

Mostra mercato nazionale radiantistica elettronica

L'evento marchigiano per tutti gli appassionati di elettronica e radiantistica. Oltre ai componenti elettronici e radioamatoriali, troverete computer, accessori per la telefonia cellulare, stampanti, CD, DVD e gadget.

Dove: Civitanova Marche (MC) ***Quando:*** 12-13 Dicembre 2009

Orari: dalle 9.00 alle 18.00 ***Organizzazione:*** E.R.F.

Info: www.erf.it

CODICE MIP 2778757

12-13 Dicembre 2009

EXPO ELETTRONICA Cremona

Expo Elettronica, il grande circuito di mostre mercato dedicate all'elettronica professionale e di consumo, fa tappa a Cremona un nuovo appuntamento dedicato al pubblico lombardo ed emiliano. La rassegna è 'l'anello mancante' fra "antiquariato tecnologico" e applicazioni "futuribili": una miriade di oggetti e applicazioni ormai indispensabili come computer, software, periferiche, home theater, telefonia fissa e mobile, accessori, ricambi, curiosità elettroniche e digitali.

Dove: Cremona Quando: 12-13 Dicembre 2009

Orari: dalle 9.00 alle 18.00 Organizzazione: Blunautilus

Info: www.expoelettronica.it

CODICE MIP 2778761

12-13 DICEMBRE 2009

Computerfest & Radioamatore

Singoli componenti, periferiche, software, personal computer ma non solo. Alla ComputerFest di Udine potrete trovare apparati radio e componentistica radioamatoriale; home entertainment (tv-sat, dvd, antenne); telefonia fissa e mobile; editoria specializzata; fai-da-te elettronico, attrezzature e strumentazioni varie.

Dove: Reggio Emilia

Quando: 12-13 dicembre 2009

Orari: dalle 9.00 alle 18.00

Organizzazione: Compendio Fiere

info: www.compendiofiere.it

CODICE MIP 2778765

19-20 Dicembre 2009

Marc Mostra Mercato

La manifestazione organizzata da Studio Fulcro con il patrocinio dell'ARI – Associazione Radioamatori Italiani – si svolgerà nel Padiglione C e offrirà una panoramica completa di attrezzature e componentistica radioamatoriale, oltre a parabole satellitari e antenne, decoder, autoradio e navigatori satellitari GPS, fotocamere e videocamere digitali, mini TV Lcd, lettori MP3, console per playstation e videogiochi, accessori per i telefoni cellulari, palmari, fax e telefonia in genere. Grandissimo risalto avrà il settore informatico con un'ampia offerta di pc, notebook, stampanti, scanner, software, monitor Lcd, CD, DVD, Usb pen drive, webcam. Inoltre, ampissima offerta per ricariche di inchiostro e di cartucce compatibili per le stampanti, mentre i cultori del fai-da-te potranno trovare pezzi di ricambio, schede video, Hard disk, apparecchiature da mettere a posto e accessori per elaborazioni speciali a costi estremamente concorrenziali.

Dove: *Genova*
Quando: *19-20 Dicembre 2009*
Orari:: *dalle 9.00 alle 18.00*
Organizzazione: *Studio Fulcro*
Info: www.studio-fulcro.it

CODICE MIP 2778759

19-20 Dicembre 2009

ELECTRONIC DAYS

Informatica, computer, cd e dvd, editoria specializzata, antenne, sistemi satellitari e ricambi, batterie, apparati per radioamatori, MP3/4 e tanti altri articoli. Parcheggio gratuito, bancomat e punto ristoro.

Dove: Biella Fiere
Quando: 19-20 Dicembre 2009
Orari: dalle 9.00 alle 18.00
Organizzazione: Italfiere
Info: www.electronicdays.it

CODICE MIP 2778763

CODICE MIP 2777740

Soluzioni switch specifiche per moduli fotocamera

Fairchild Semiconductor offre ai progettisti di smartphone 3G, cellulari, set-top box, videocamere di sicurezza e notebook un vasto portafoglio di switch per fotocamere che risolvono le complesse sfide progettuali tipiche di queste applicazioni. Le soluzioni dedicate di Fairchild si caratterizzano per: isolamento totale, capacità di gestire interfacce multiple, flessibilità progettuale, emissioni EMI contenute, On capacitance complessiva inferiore. Fairchild Semiconductor fornisce soluzioni all'avanguardia per sistemi portatili. I suoi switch analogici, audio, video, camera e USB rivestono un'importanza critica nel ridurre il numero di circuiti, eliminare i connettori superflui, semplificare il design e risparmiare spazio prezioso all'interno degli apparecchi.

CODICE MIP 2775124

MIXER DOWNCONVERTING PER STAZIONI BASE
wireless 3G/4G

Linear Technology annuncia la famiglia LTC554x composta da quattro nuovi mixer downconverter con ampio range dinamico in grado di convertire intervalli di frequenza tipici delle infrastrutture wireless, da 700 MHz a 4 GHz. I mixer della gamma LTC554x offrono un IIP3 (input third-order intercept) di 26 dBm, cifra di rumore di soli 9-10 dB e un elevato guadagno di conversione di 8 dB, garantendo così prestazioni eccellenti nel range dinamico sia per i ricevitori principali che per i ricevitori con pre-distorsione digitale. La famiglia di mixer LTC554x fornisce anche funzionalità avanzate per mantenere bassa la cifra di rumore in presenza di forti interferenze di blocco e migliorare al tempo stesso la sensibilità e l'affidabilità del ricevitore. Diversamente da altri mixer della stessa categoria, i dispositivi della famiglia LTC554x funzionano con una singola alimentazione da 3,3 V senza pregiudicare le prestazioni e consumano il 34% in meno rispetto alla principale soluzione concorrente.

CODICE MIP 2776004

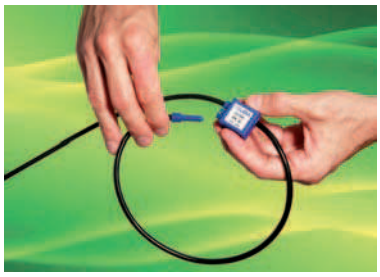
CODICE MIP 2755611

PROGETTO VIACTORS:
i robot simili all'uomo

Il 25 ottobre, a Palazzo Ducale di Genova, nell'ambito del Festival della Scienza, è stato presentato il progetto europeo VIACTORS, per la costruzione di robot simili all'uomo. Una volta erano colossi di metallo, o cervelloni con il corpo cubico, immensi elaboratori di dati. Niente di tutto questo per la moderna robotica: intelligenza e capacità di compiere determinate prestazioni sembrano dipendere sempre di più dal tipo di struttura fisica di cui dotiamo gli automi. Proprio come accade negli esseri umani, in cui alcune capacità cognitive e di movimento dipendono dalla struttura neuromuscolare. Con l'obiettivo di correlare l'interazione tra la struttura, la morfologia, e la capacità di compiere determinate prestazioni è nato VIACTORS (Variable Impedance ACTtuation systems embodying advanced interaction behaviORS), un progetto europeo che vede coinvolte anche il Centro per l'Automatica, la Robotica e la Bioingegneria "Enrico Piaggio" dell'Università di Pisa e l'Istituto Italiano di Tecnologia di Genova.

CODICE MIP 2775309

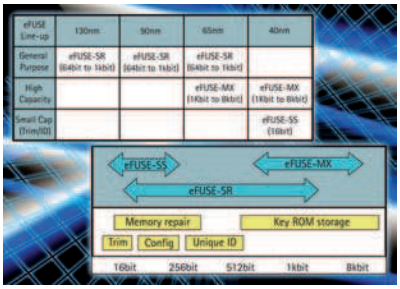
Nuovi sensori a bobina
di Rogowski da LEM



LEM presenta la prima famiglia di sensori di corrente in AC basata sulla tecnologia Perfect Loop, con brevetto in attesa di approvazione. Con calibrazione, i nuovi RT raggiungono la precisione in termini assoluti di oltre 0,65%, incluso l'errore di posizione, rendendo questi sensori le prime bobine di Rogowski apribili adatte per l'uso nei dispositivi di potenza di Classe 1. La nuova serie RT ovvia al problema di asimmetria derivante dalla discontinuità in corrispondenza dell'apertura del sensore, che è intrinseco nelle bobine convenzionali di Rogowski apribili. Una struttura imperfetta dell'avvolgimento induce una geometria sbilanciata e aumenta la sensibilità verso la posizione del conduttore da misurare che passa all'interno del sensore, o in prossimità dei cavi elettrici esterni. Mentre altri produttori utilizzano sistemi complessi e costosi di chiusura per minimizzare gli effetti della discontinuità, gli ingegneri LEM hanno inventato una tecnologia unica ad accoppiamento magnetico, con brevetto in fase di approvazione, che consente di ottenere un'estensione perfetta del flusso magnetico in corrispondenza dell'apertura dell'anello e che compensa l'asimmetria della bobina.

CODICE MIP 2775340

NUOVA TECNOLOGIA EFUSE DI TOSHIBA



Il centro ricerche ASIC & Foundry di Toshiba Electronics Europe (TEE) ha annunciato che da oggi gli utenti europei di dispositivi ASIC e SoC (System On Chip) possono avvalersi della tecnologia eFUSE (electrical programmable fuse) nello sviluppo di circuiti integrati basati su processi Toshiba. La tecnologia eFUSE utilizza antifusibili PMOS a ossido di gate, programmabili elettricamente, e offre una soluzione versatile, economica e facile da usare per la regolazione dei parametri di prestazione, il ripristino delle memorie e l'aggiornamento delle configurazioni e dei dati di versione. eFUSE offre inoltre funzioni di memorizzazione e tracciamento di 'chip ID' e la possibilità di memorizzare tramite chiave ROM i dati delle trasmissioni sicure.

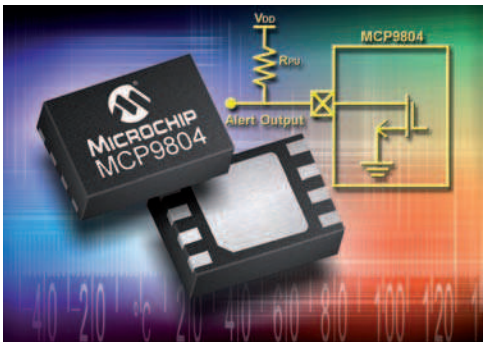
CODICE MIP 2776002

LA PRIMA MEMORIA NAND FLASH DA 64 GIGABIT X4

SanDisk, leader mondiale nella produzione di schede di memoria flash, annuncia di aver iniziato la distribuzione di memorie flash basate sulla propria tecnologia avanzata di memoria flash X4, che contiene 4 bit di dati in ogni cella di memoria, il doppio rispetto alle celle dei tradizionali chip di memoria NAND MLC (Multi-Level Cell, da 2-bits-per-cell). Basato su una tecnologia di processo a 43 nanometri (nm), il chip NAND flash da 64 Gigabit (Gb) è il dispositivo di memoria a die singolo con più alta densità in assoluto a entrare in fase di produzione. Lo sviluppo e la commercializzazione della tecnologia X4 rappresenta un traguardo importante nel settore dell'archiviazione flash e la distribuzione della tecnologia a 4 bit per cella è un'evoluzione necessaria per il mercato.

CODICE MIP 2774979

NUOVO SENSORE MCP9804 **di Microchip**



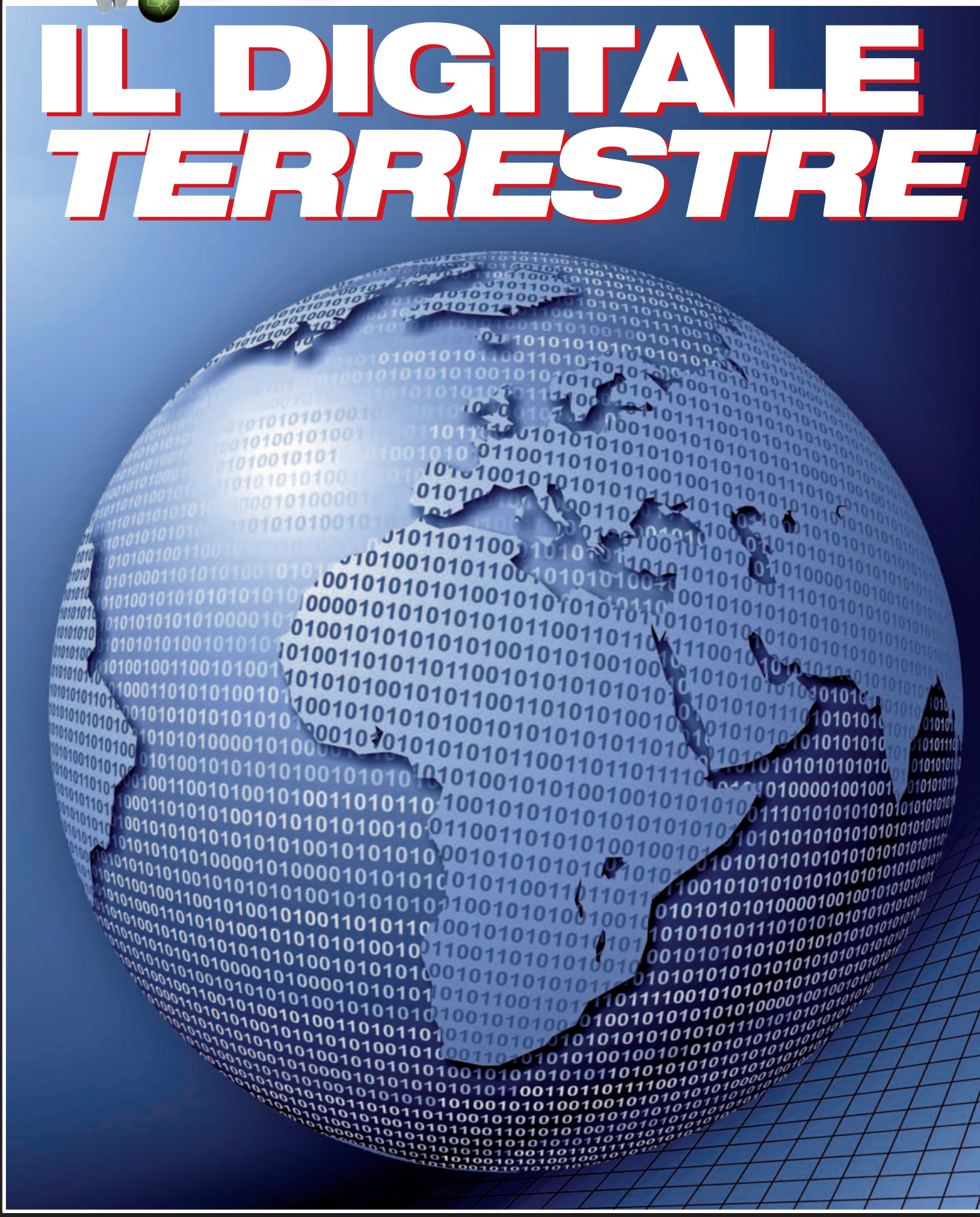
Microchip annuncia il nuovo sensore di temperatura MCP9804. Il sensore offre una precisione tipica di +0.25°C o di +/- 1°C da -40 a +125°C, e un consumo statico di soli 200µA (typ). Disponibile in package miniaturizzati MSOP a 8-pin e 2mm x 3mm DFN, questo dispositivo I2C™ permette di ridurre gli ingombri su scheda e garantisce autonomie di batteria prolungate nelle applicazioni industriali, automotive e consumer.

Esempi applicativi del sensore di temperatura MCP9804 comprendono i freezer industriali (i quali richiedono un'elevata precisione a temperature da -20°C a+45°C), i dispositivi elettronici consumer come i personal computer (i quali richiedono elevata precisione a +85°C) e i sistemi automotive come i misuratori di temperatura del propulsore, che presuppongono alte precisioni fino a +125°C.

CODICE MIP 2774370

di GIUSEPPE DE GENNARO

IL DIGITALE TERRESTRE



Un cambiamento radicale di uno degli oggetti più utilizzati e maggiormente diffusi nel mondo. In questo articolo cercheremo di scoprire gli aspetti tecnici della televisione digitale, partendo dalla codifica dei segnali audio/video, passando per la modulazione, la trasmissione radio terrestre e la ricezione su un set-top-box

Forse non tutti sanno che stiamo vivendo, proprio in questi mesi, una delle più grandi rivoluzioni tecnologiche dell'era moderna relativa all'oggetto elettronico forse più diffuso e usato dalla quasi totalità della popolazione mondiale: la televisione. L'invenzione e la diffusione della ormai onnipresente "scatola animata" avevano già dato inizio, a partire dalla metà del secolo scorso, a una grande rivoluzione tecnologica e una, forse anche più grande, sociologica. Per quasi sessanta anni, ignorando il passaggio dalla televisione in bianco e nero a quella a colori, il sistema è rimasto pressoché identico.
Oggi, però, è messo completamente in discussione e, nel giro di qualche anno, almeno nelle previsioni dei governi, sarà completamente abbandonato, entrando a tutti gli effetti nella storia. Il nuovo sistema porterà con sé numerosi vantaggi e servizi aggiuntivi che solo una tecnologia digitale può fornire: un maggior numero di canali televisivi, servizi interattivi, applicazioni e navigazione internet direttamente sul televisore, migliore qualità audio/video, alta definizione, pay-tv.
In questo articolo, dopo una breve nostalgica panoramica sul "vecchio" sistema televisivo, saranno analizzati gli aspetti tecnici del digitale terrestre, lasciando ad altri i risvolti politici e sociali, allo stesso modo importanti.

LA TELEVISIONE ANALOGICA

Così come si intuisce dal nome, il segnale ricevuto ed elaborato dalla televisione storica è completamente analogico (e, molto spesso, continua a rimanere analogico anche negli ambienti di regia, nello studio, nei siti di trasmissione delle emittenti).

Banda base

Prima di essere trasmesso via radio, il segnale televisivo ha un contenuto spettrale di qualche MHz e viene quindi denominato segnale in "banda base" o a "bassa frequenza". Esistono diversi tipi di segnali video analogici in banda base, il più importante, relativamente alla trasmissione televisiva, è il cosiddetto video composito. Il segnale televisivo contiene una sequenza di immagini statiche (quadri o tecnicamente *raster*) che vengono visualizzate, una dopo l'altra, ad alta velocità, in modo che l'occhio umano abbia un senso di fluidità del movimento. Nel video composito, la sequenza dei quadri nel tempo è scandita da un segnale di sincronismo (livello di tensione convenzionale di -300 mV) detto, appunto, sincronismo di quadro.
Ogni quadro rappresenta un'immagine statica, una sorta di fotografia in un istante ben preciso. Ignorando la questione del colore e considerando una immagine in bianco e nero, un quadro è una figura bidimensionale caratterizzata da un livello di luminosità variabile da punto a punto. Per descriverla tutta viene effettuata una scansione *raster* (*raster scan*): si parte dal punto in alto a sinistra e, muovendosi verso destra, si descrive, mediante un livello di tensione convenzionale variabile da un minimo di 0 mV a un massimo di 700 mV, la luminosità della prima linea. Dopo la scansione della prima riga, il processo riprende dalla linea immediatamente sotto e così via (questa scansione è simile a quella fatta dai nostri occhi quando leggono un testo su un foglio di carta). Nel video composito, il segnale relativo a una riga è separato da quello della riga successiva da un altro sincronismo, detto appunto di riga. I sincronismi sono importanti, poiché il ricevitore (cioè il televisore) che riceve questo segnale dovrà essere in grado di riprodurre la stessa immagine trasmessa e quindi dovrà visualizzare la prima riga esattamente in alto, la seconda subito dopo, ecc. Per fare ciò, il ricevitore andrà alla ricerca del sincronismo di quadro (con parole più tecniche, si aggancerà ai sincronismi di quadro), dal quale partirà per tracciare l'immagine sullo schermo. Allo stesso modo il ricevitore si aggancerà al sincronismo di linea per passare alla riga successiva.
In **figura 1** è mostrata una schematizzazione del segnale video analogico nella parte relativa a una sola linea, dove si evidenziano il sincronismo di riga e il segnale di luminosità.
In questa brevissima e ovviamente limitata trattazione non sono citati alcuni argomenti fondamentali come l'interlacciamento (i quadri in realtà sono divisi in due semiquadri), i tempi e le frequenze, il colore, l'audio (mono, stereo e dual-sound), il multistandard (PAL, NTSC, ecc), il teletext, le righe di test, ecc.

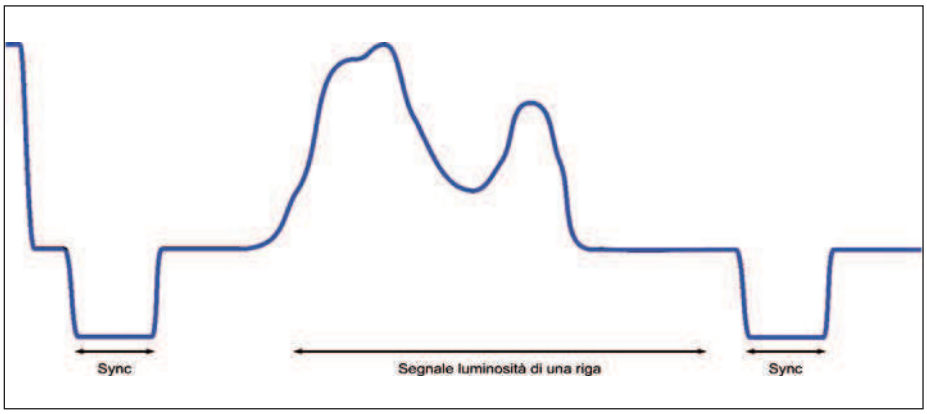


Figura 1: esempio di un segnale video nel tempo.

Come conclusione è opportuno sottolineare come il segnale trasmesso dalle emittenti è una sorta di "mano virtuale" che, dal sito di trasmissione, guida i pennelli di tutti i televisori sintonizzati su quella frequenza durante il disegno delle righe, delle immagini e quindi dell'intero video.
Questo concetto di "guida" sarà abbandonato grazie alla tecnologia digitale.

IL DIGITALE TERRESTRE

Quando si parla di televisione digitale terrestre (più amichevolmente, digitale terrestre o DTT), ci si riferisce a una tecnologia televisiva che tratta tutti i segnali (principalmente audio/video, ma anche dati) in digitale, quindi mediante due soli valori numerici: 1 e 0.
Il mondo che ci circonda è analogico (sfido chiunque ad aver mai visto il colore o sentito il rumore di un bit) e quindi le immagini e il suono vengono "da qualche parte" convertite dall'analogico al digitale. Questi 1 e 0 in fila (immaginiamoli in marcia, uno dopo l'altro, perfettamente allineati e coperti) attraversano cavi, circuiti elettronici, memorie, processori e, alla fine, arrivano al sito di trasmissione dove vengono lanciati nell'etere, uno dopo l'altro. Per fortuna questi poveri bit in volo vengono raccolti dalla mano amica dell'antenna posta sul nostro tetto e incanalati ancora una volta all'interno di cavi, circuiti stampati, microprocessori fino a essere, finalmente, convertiti nuovamente in analogico, nel colore rosso di un vestito da soubrette o nel tono melodico di un violino durante un concerto alla Scala. È di questa "saga dei bit" che vogliamo scrivere, abbandonando questa descrizione un po' troppo romanzata.

Il mondo è bello perché è vario

Quando ci riferiamo al digitale terrestre, probabilmente lo immaginiamo come un unico sistema valido in tutto il mondo. In realtà, come per tutti gli standard internazionali, ci sono vari digitali terrestri: quello europeo, quello americano, quello cinese, quello giapponese, ecc. Non entreremo nel merito dei motivi, soprattutto economici, che portano alla proliferazione degli standard, ma ci limitiamo soltanto a nominarli:

- il digitale terrestre europeo, denominato DVB-T (Digital Video Broadcasting – Terrestrial);
- il digitale terrestre americano ATSC (Advanced Television Systems Committee);
- il digitale terrestre cinese DTMB (Digital Terrestrial Multimedia Broadcast);
- il digitale terrestre giapponese/brasiliano ISDB (Integrated Services Digital Broadcasting).

Nel complesso sistema digitale, molte parti di questi standard sono comuni o, quantomeno, molto simili (come detto, non c'erano reali motivi tecnici per far nascere standard diversi). In questo articolo ci concentreremo sul digitale terrestre che ci riguarda in prima persona, il DVB-T.

Il consorzio DVB

L'acronimo DVB (Digital Video Broadcasting) è sia il nome di un consorzio internazionale di aziende ed enti di regolamentazione (www.dvb.org) che si occupa della stesura dei documenti tecnici della nuova televisione digitale e della loro diffusione principalmente in Europa, ma anche nel resto del mondo, sia il nome dello standard e della tecnologia in sé. Giusto

per dare un'idea del calibro dei membri di questo consorzio, citiamo (in rigoroso ordine alfabetico) AMD, BBC, Cisco Systems, Deutsche Telekom, Fracarro, Freescale, Fujitsu, HP, Humax, Intel, LG, Mediaset, Microsoft, Ministero delle Comunicazioni, Motorola, Nokia, Panasonic, Philips, RAI, Samsung e Telecom Italia.

Non solo terrestre

Un sistema televisivo completo riguarda non soltanto la modalità di trasmissione (nel nostro caso "terrestre"), ma anche gli aspetti relativi al segnale audio/video in "banda base". È per questo che il DVB è principalmente una piattaforma di standard digitali televisivi che hanno una base comune e si dividono soltanto nella modalità di trasmissione: terrestre (DVB-T), via cavo (DVB-C), satellitare (DVB-S), mobile (DVB-H). Molti argomenti che affronteremo nel seguito, quindi, soprattutto quelli che hanno a che fare con il trattamento dei segnali audio/video in digitale, non devono essere associati al solo "digitale terrestre", ma anche ad altri tipi di televisione digitale.

CODIFICA AUDIO/VIDEO

Se la televisione deve essere digitale, tale deve essere anche il segnale audio/video, principale servizio offerto dal sistema. A differenza della trasmissione che per anni è rimasta analogica (e tuttora lo è), negli studi delle emittenti televisive le telecamere, i mixer, gli elaboratori audio/video sono diventati già da tempo digitali nella maggior parte dei casi. La conversione analogico/digitale è, quindi, sempre più spesso, una delle prime fasi della creazione di un video, e avviene già all'interno delle telecamere. In alcuni casi il segnale televisivo è presente in formato analogico (si pensi ai vecchi film su pellicola o a vecchie telecamere analogiche ormai in disuso) e dovrà essere necessariamente convertito in digitale prima di poter essere trasmesso. Questa conversione analogico/digitale può avvenire con diverse apparecchiature che permettono di eliminare il più possibile i difetti dell'analogico.

Audio/video digitale

Il segnale video, sin dai tempi della televisione analogica, non è altro che una sequenza di immagini statiche, dette *frame* (il termine *raster* visto precedentemente è legato maggiormente alla tecnologia analogica), visualizzate velocemente una dopo l'altra in modo da ingannare l'occhio e fargli percepire una certa fluidità di movimento.
In digitale, il frame viene diviso in una matrice di punti, detti pixel, in orizzontale e verticale. Il numero dei pixel indica la risoluzione dell'immagine e, più in generale, del video. Ogni pixel viene trasformato in tre numeri che indicano la quantità di colore rosso, verde e blu, relativa a quel punto, nell'immagine originale. Tipicamente ogni colore viene rappresentato da 10 bit (standard SDI utilizzato nelle telecamere digitali professionali), quindi da 1024 livelli di colore rosso, verde e blu. Un'immagine con una risoluzione pari a 720x576 (tipica risoluzione video utilizzata nei DVD e anche nell'interfaccia SDI) occupa quindi 3x720x576x10 = 12441600 bit.
Considerando 25 *frame* al secondo, si ottiene un *bitrate* (numero di bit al secondo) di circa 12441600x25 ≈ 311Mbit/s!! È ovvio che si tratta di una velocità molto alta (si pensi che la più veloce ADSL commerciata in Italia è di 20 Mb/s), praticamente impossibile da trasmettere via etere. E, pur se in modo più limitato, non è stato considerato lo spazio occupato dall'audio.
Anche se apparentemente sembra che il digitale ci abbia portato in un vicolo cieco, in realtà, proprio tecniche digitali ci permettono di abbassare notevolmente i requisiti di bitrate richiesti nella trasmissione video. Molti di voi avranno già capito che stiamo parlando della tecnica di compressione.

Compressione

La compressione è un procedimento numerico che permette di ridurre la dimensione (nel caso di memorizzazione) o il *bitrate* (nel caso di trasmissione) di un'informazione, quali immagini, audio, video, dati. Le tecniche di compressione si dividono in due grandi gruppi: quelle senza perdita (*lossless compression*) e quelle con perdita (*lossy compression*).
Le prime sono ampiamente utilizzate nel mondo informatico per minimizzare la dimensione di file (eseguibili, archivi, database, documenti, ecc). Il procedimento inverso della compressione (decompressione) deve dare come risultato lo stesso file di partenza, altrimenti esso non potrà essere correttamente interpretato dal computer.
Le compressioni con perdita, invece, sono ampiamente utilizzate nell'ambito multimediale per diminuire lo spazio di memorizzazione e la banda di trasmissione (cioè il bitrate) di informazioni audio e video. Infatti, in questi casi, è sufficiente "ingannare" l'occhio e l'orecchio umano (i veri destinatari di queste informazioni), senza pretendere di ottenere la stessa informazione originale. Si potrebbe pensare che una compressione *lossy* possa portare a una qualità molto bassa (molti pensano di avere un orecchio così fine da poter notare la differenza di un bit in un flusso audio digitale). In realtà, il successo dei lettori MP3, delle macchine fotografiche digitali compatte, dei DVD e proprio del nostro caro digitale terrestre, dimostra esattamente il contrario. Ci limitiamo qui a citare i tre standard più diffusi di compressione *lossy*, ma, in realtà, ne esistono molti altri, più o meno performanti: MP3 per l'audio, JPEG per le immagini, MPEG per il video (in realtà, l'MP3 fa parte del più grande standard MPEG).
Una delle prime decisioni prese in seno al consorzio DVB è stata quella di scegliere, come cuore del sistema, la codifica MPEG-2, sia in termini di compressione che di trasporto (come si vedrà in seguito). In particolare, si è pensato di utilizzare comunemente l'MPEG-2 layer II (il più noto MP3 è, in realtà, MPEG-1 layer III) e l'MPEG-2 Main Profile @ Main Level per la compressione audio e video rispettivamente. Purtroppo queste tecniche di compressione audio/video sono molto sofisticate e meriterebbero molto più spazio. Qui di seguito presentiamo sommariamente, senza pretese di completezza, le principali caratteristiche delle tecniche di compressione utilizzate nel DVB.

MPEG-2 Layer II

La principale tecnica di compressione utilizzata nell'MPEG-1 (adottata, in seguito, dall'MPEG-2 praticamente inalterata) è quella di trasformare il segnale audio (variabile nel tempo) nel dominio della frequenza, sfruttando un filtro a 32 sotto-bande. In pratica, per ogni sotto-banda viene calcolata la relativa energia del segnale. Successivamente viene applicato un modello psicoacustico dell'orecchio umano che permette di individuare le informazioni inutili che, anche se trasmesse inalterate, non verrebbero comunque udite. Un tipico esempio è

quello di due toni a due diverse frequenze, uno molto più alto (in termini di volume) dell'altro. Il tono più basso non verrà sentito dall'orecchio e quindi la sua informazione può essere eliminata. Naturalmente, quante più informazioni vengono scartate, tanto più diverso sarà il segnale audio decompresso rispetto a quello originale. Lo standard permette di scegliere un compromesso tra qualità e quantità, utilizzando il bitrate che meglio si adatta all'applicazione. Sono previsti molteplici bitrate: 32, 48, 56, 64, 80, 96, 112, 128, 160, 192, 224, 256, 320, 384 kbit/s.

MPEG-2 Main Profile @ Main Level

La compressione video MPEG-2 è basata su profili e livelli. La combinazione maggiormente utilizzata nel DVB è quella denominata Main Profile @ Main Level (MP@ML). Una prima compressione viene effettuata descrivendo i pixel dei frame mediante le informazioni di luminanza e crominanza (codifica denominata YCbCr), anziché quelle sui colori rosso, verde e blu. Questa "strana" codifica permette di sfruttare una limitazione dell'occhio umano che riesce ad apprezzare molto meglio i dettagli di luminanza, piuttosto che quelli di crominanza. Una semplice riduzione di bitrate si ottiene prendendo, per ogni area di 2x2 pixel, soltanto un valore di crominanza Cb e uno Cr (sottocampionamento della crominanza secondo lo schema 4:2:0). Il bitrate complessivo risulta dimezzato (4 valori Y, 1 Cb e 1 Cr) rispetto a nessun sottocampionamento di crominanza (4 valori Y, 4 Cb e 4 Cr). In **figura 2** è illustrato graficamente questo sistema: ogni quadrato rappresenta un pixel, l'area di 4x4 pixel è indicata con un bordo più spesso, il campione di crominanza colore è rappresentato con un cerchietto. Ogni frame del video viene classificato, secondo uno schema abbastanza personalizzabile, in I-frame, P-frame o B-frame.

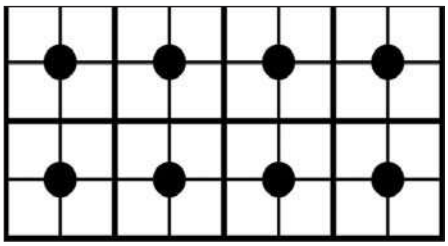


Figura 2: schema di sottocampionamento dell'informazione di crominanza.

Un frame di tipo I (*intra-coded*) viene compresso considerando soltanto quel frame. Mediante operazioni matematiche di trasformazione nel dominio della frequenza (anche nello spazio, e non solo nel tempo, si può parlare di frequenze), vengono eliminate informazioni che l'occhio umano non potrebbe vedere, per esempio cambiamenti di luminanza/crominanza in spazi molto piccoli. La caratteristica fondamentale di un I-frame è quella di poter essere sempre decodificato senza conoscere nulla del frame precedente o successivo.

Al contrario, un P-frame (*predicted-frame*) viene compresso sfruttando le sue similitudini con il frame immediatamente precedente (per esempio, in un panorama il cielo rimane pressoché nella stessa posizione ed è quindi inutile ritrasmettere la stessa informazione per ogni frame). È bene notare che, se il frame precedente viene trasmesso in modo errato, altrettanto errata sarà la decodifica del P-frame che da questi dipende. Il vantaggio è che un P-frame può essere compresso con un numero minore di bit rispetto a un I-frame.

Infine, il B-frame (*bidirectional-frame*) viene codificato sfruttando le sue similitudini sia con il frame precedente che con quello successivo. Anche in questo caso, se la trasmissione dei frame di riferimento è errata, il B-frame viene a sua volta decodificato in modo errato. Per contro, un B-frame può essere compresso maggiormente rispetto a un P-frame (e quindi a un I-frame).

Il compressore utilizzerà una sequenza ben precisa di frame I, P e B che viene denominata GOP (*Group Of Pictures*). È bene notare che è necessario partire sempre con un I-frame che farà anche da riferimento per i successivi frame predittivi (P e B). Siccome gli errori di un frame si propagano in cascata in tutti gli altri frame P e B, è bene che la sequenza GOP non sia molto lunga e termini con un altro I-frame, che farà da riferimento per la successiva sequenza. Una possibile GOP può essere: IBBPBBPBBPBB(I). Lo standard video MPEG-2 definisce i cosiddetti livelli e profili che specificano altri dettagli relativi alla compressione video. Nell'MP@ML utilizzato spesso nel DVB, la risoluzione video è 720x576 e il sottocampionamento colore è il 4:2:0. Alla fine

ELETTRONICA E NON SOLO

Tutto quello che ti serve

480.000 prodotti da
3.500 marchi leader

Quando ti serve

Consegna in 24 ore

www.farnell.com/it

- Gamma di componenti da produttori leader
- Disponibilità per consegne in 24 ore
- Le tecnologie all'avanguardia, disponibili in anteprima
- Nessun minimo ordinabile
- Servizio affidabile e garantito

Farnell

Progetta con i migliori

A Premier Farnell Company

CODICE MIP 2772103

di questi procedimenti di compressione, si ottiene un flusso dati che tipicamente varia da 2 a 8 Mbit/s, a seconda della qualità che si vuole ottenere e dal tipo di programma (un filmato molto veloce, come sport o film d'azione, richiederà un bitrate maggiore).

FLUSSO DI BIT

A differenza di un sistema informatico dove i dati devono essere tipicamente memorizzati su un supporto per una successiva consultazione, in un sistema di diffusione televisiva, l'audio e il video digitali devono essere trasportati da un punto a un altro, da un apparato a un altro, da un circuito a un altro. Il consorzio DVB ha scelto, anche per questo scopo, lo standard MPEG-2 che già aveva definito due tipi di flussi (in inglese, *streams*): program *stream* e *transport stream*.

Program stream

Idealmente un codificatore MPEG-2 è formato da un compressore video, un compresso audio e un multiplexer (si veda la **figura 3**). Ogni compressore produce in uscita, con un bitrate diverso, una sequenza di pacchetti contenenti, rispettivamente, informazioni su una parte del video e una parte dell'audio. Questo flusso di pacchetti viene indicato come PES (*Packetized Elementary Stream*). I due flussi elementari video e audio, ognuno con un bitrate diverso, vengono multiplexati per ottenere un unico flusso denominato PS (*Program Stream*). Oltre ai pacchetti PES-Video e PES-Audio, il codificatore aggiunge altre informazioni generali utili al decodificatore, prima fra tutte un segnale di temporizzazione che permette all'audio e al video un perfetto sincronismo. In pratica, un PS contiene tutti i dati relativi a un programma video.

Transport stream

Il *program stream* non è utilizzato nello standard DVB, poiché è più orientato alla memorizzazione (è infatti utilizzato nei DVD-Video) o alla trasmissione su canali a bassa probabilità di errore (cosa che non è nel caso di trasmissioni radio terrestri). Al posto del *program stream*, viene utilizzato il cosiddetto *Transport Stream* (TS) che, come si evince dal nome, è più orientato al trasporto, cioè alla trasmissione. Caratteristica fondamentale di un TS è la sua struttura a pacchetti di lunghezza fissa (pacchetti di trasporto) da 188 o 204 byte. I flussi PES vengono miscelati insieme e trasformati in un unico TS direttamente nel codificatore MPEG. Un'altra caratteristica del TS, utilizzata ampiamente in ambito DVB, è la possibilità di multiplexare insieme due flussi TS in un unico TS risultante (naturalmente con un bitrate maggiore). In pratica, è con questa tecnica che è possibile ottenere un bouquet (cioè un insieme) di programmi televisivi che formano un tutt'uno, che è possibile trasmettere nello stesso canale a radiofrequenza.
Come si può vedere, le tecniche digitali ci portano a un enorme vantaggio (uno fra

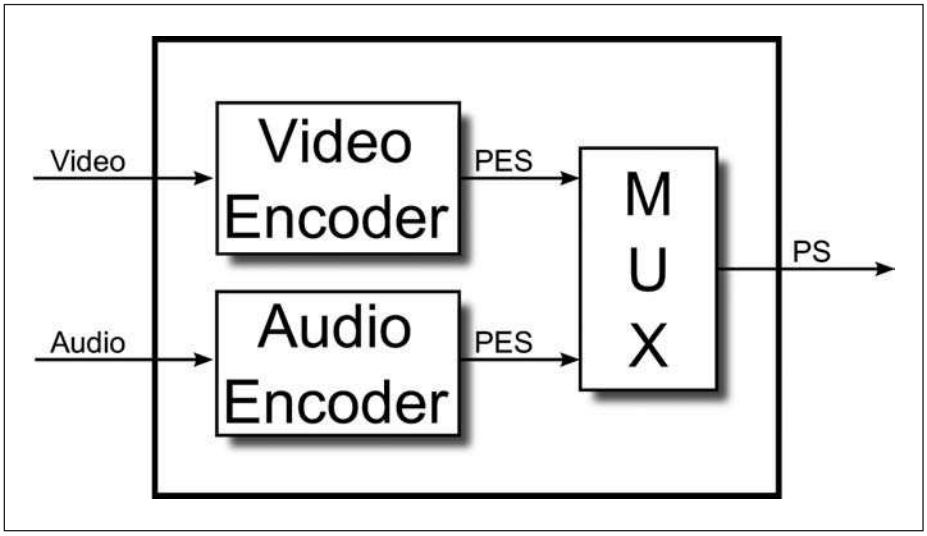


Figura 3: schema di principio di funzionamento di un codificatore MPEG-2.

tanti) rispetto alla televisione analogica. In un singolo canale televisivo è ora possibile trasmettere più programmi televisivi, con uno sfruttamento decisamente migliore della banda limitata a disposizione delle emittenti. Inutile scendere nei dettagli della composizione del transport stream, analizzando la struttura di un pacchetto e dei possibili contenuti che possono essere di vario tipo, tutti classificati mediante un codice identificativo detto PID (*Packet IDentifier*). Oltre ai pacchetti audio, video e di temporizzazione già visti, sono presenti anche (non tutti sono obbligatori):

- PAT e PMT: strutture dati che consentono al decodificatore TS di sapere quali e quanti programmi sono presenti e come poterli estrarre;
- NIT: informazione obbligatoria che contiene alcune informazioni sulla trasmissione (frequenza, numero di postazione, ecc.) e sulla rete (tramite un nome e un identificativo) che genera il TS;
- CAT: informazioni utilizzate per la decodifica di eventuali programmi criptati;
- SDT: struttura dati contenente, tra l'altro, il nome dei programmi presenti nel TS;
- EIT: informazioni riguardanti il programma attualmente in corso ed eventualmente i programmi successivi.

Interfacce elettriche

È bello parlare di flussi di bit ad alta velocità, ma non perdiamo di vista l'aspetto tecnico. Come vengono trasmessi, per esempio da un circuito a un altro (all'interno di un apparato) o da un apparato all'altro (per esempio da un encoder MPEG a un multiplexer) da un punto di vista elettrico? Lo standard DVB prevede due tipi di interfacce, denominate SPI e ASI.
L'interfaccia SPI (*Syncronous Parallel Interface*) è di tipo parallelo, formata da 11 segnali LVDS (*Low-Voltage Differential Signal*): 8 bit di dati, un segnale di clock, uno di sincronismo attivo sul primo byte di ogni pacchetto da 188/204 byte, uno di validità dei dati attivo per indicare la trasmissione di informazioni valide.
Si è scelto di utilizzare segnali differenziali LVDS, poiché essi permettono di raggiungere distanze maggiori a frequenze elevate. È anche previsto un connettore standard D-SUB da 25 poli su cui sono presenti questi 11 segnali differenziali (ricordiamo che per ogni segnale differenziale sono utilizzati due poli). In realtà, quest'interfaccia viene spesso utilizzata all'interno di apparati, poiché l'LVDS è facilmente interfacciabile direttamente con integrati, come FPGA. In alcuni casi, per minimizzare i costi e per distanze molto brevi (per esempio, da chip a chip su uno stesso circuito), gli 11 segnali dell'SPI vengono trasmessi in LVTTL (*Low-Voltage TTL*) anziché LVDS.
L'altra interfaccia standard, utilizzata nella interconnessione di apparati anche a distanza elevata, è il cosiddetto ASI (*Asyncronous Serial Interface*). Essa è di tipo seriale e trasmessa su un cavo coassiale a 75 Ω, con connettori terminali di tipo BNC. La velocità di riferimento è di 270 Mbit/s e i dati sono trasmessi effettivamente a questa velocità, ma, poiché un TS ha una velocità molto variabile dipendente dalla configurazione, vengono trasmessi byte "nulli" di *stuffing* quando necessario.

TRASMISSIONE TERRESTRE

Tutti i concetti visti fino a questo punto sono validi in tutta la piattaforma DVB, indipendentemente dal tipo di trasmissione dal broadcaster (l'emittente) all'utente finale. Da questo paragrafo in poi, tutto quello che verrà detto sarà valido soltanto per la modalità di trasmissione terrestre, cioè per il DVB-T.
Scopo della trasmissione terrestre è quello di inviare un TS dalla stazione trasmittente alle antenne terrestri (posizionate normalmente sui tetti dei palazzi) utilizzando canali radiofrequenza già allocati per la televisione analogica (per evitare incompatibilità nelle antenne e nella coesistenza dei due servizi) nelle bande III (174-230 MHz), IV e V (470-858 MHz) di larghezza 7 o 8 MHz.
Il periodo di transizione dalla televisione analogica alla televisione completamente digitale deve essere necessariamente molto lungo (dell'ordine di decine di anni), ed è quindi fondamentale prevedere la coesistenza tra le due tipologie di servizi. È per questo che nessuno si è sognato di modificare le bande e la canalizzazione già utilizzata nella televisione analogica. Inoltre, la tecnica di trasmissione digitale DVB-T deve poter convivere con la trasmissione analogica, anche su canali adiacenti.
La tecnica adottata dal consorzio DVB è la modulazione OFDM (*Orthogonal Frequency Division Multiplexing*).

OFDM

La modulazione digitale OFDM ad alto bitrate è ormai ampiamente utilizzata in numerose applicazioni che si sono dimostrate vincenti sul mercato, come l'ADSL, il Wi-Fi e, appunto, il DVB-T. È una modulazione che può essere usata sia su cavo (come nell'ADSL), sia senza fili (Wi-Fi e DVB-T). Il principio vincente dell'OFDM è quello di utilizzare, al posto di una singola portante modulata in modo digitale ad alta velocità, moltissime portanti vicine (anche nell'ordine delle migliaia) modulate a bassi bitrate, mantenendo complessivamente una velocità elevata. È come se il nostro flusso con un certo bitrate B venisse diviso tra N portanti a diversa frequenza, trasmesso in modo indipendente a un bitrate più basso (B/N), demodulato da N ricevitori e riunificato in un unico stream con bitrate B.
Il vantaggio di "sminuzzare" il flusso digitale d'ingresso in mille portanti deriva dal fatto che è possibile far fronte più facilmente a disturbi molto comuni sul canale, come:

- attenuazioni (*fading*) in banda dovute, per esempio, a sfortunate conformazioni del territorio;
- effetto *multi-path*, dovuto alla presenza di un ostacolo (quale un traliccio o un palazzo) che riflette il segnale, sommandolo, in forma attenuata e ritardata, al segnale principale (sicuramente tutti voi avrete visto almeno una volta sul televisiore una trasmissione con il cosiddetto effetto *ghosting*, cioè con una doppia immagine di cui una prevalente e l'altra sovrapposta).

Chi conosce la composizione di un tipico ricevitore starà sicuramente sgranando gli occhi, pensando alla complessità di un OFDM: ricevere N portanti è come avere N (migliaia) ricevitori, con N filtri di canale, N demodulatori, ecc. È ovvio che, se così fosse, l'OFDM non avrebbe fatto così tanta strada. La fattibilità a costi relativamente ridotti di un modulatore e demodulatore OFDM risiede in quella prima lettera O, che sta per *Orthogonality* (ortogonalità). Le N

portanti di una modulazione OFDM devono essere ortogonali, cioè deve essere possibile, per il ricevitore, separare il segnale delle varie portanti senza interferenze. Questo sarebbe possibile distanziando sufficientemente le portanti in modo da filtrare il canale di ognuna, ma richiederebbe una larghezza di banda complessiva troppo grande. Ci viene in aiuto, come spesso accade, la matematica e, in particolare, l'elaborazione numerica dei segnali. Due portanti modulate con una frequenza di simbolo Fs, sono ortogonali se distano tra di loro k x Fs, dove k è un numero intero. Quindi, se utilizzo una portante a 600 MHz modulata a 1000 simboli/secondo (ogni simbolo può convogliare due o più bit a seconda dello schema di modulazione digitale scelto), potrò utilizzare anche un'altra portante a 600 MHz + 1 kHz. E un'altra ancora a 600 MHz + 2 kHz, ecc. I soliti scettici si chiederanno come potrà un ricevitore, seppur ideale, discriminare due portanti a 600 MHz separate di soli 1 kHz. Naturalmente ciò non è fattibile filtrando il canale di ogni portante che, tra l'altro, si sovrappone con quella adiacente, ma utilizzando un meraviglioso strumento matematico dal nome FFT (*Fast Fourier Transform*). Con questo algoritmo estremamente veloce ed efficiente, implementato facilmente nei moderni DSP che hanno una sufficiente potenza di calcolo, è possibile modulare (lato trasmettitore) e demodulare (lato ricevitore) le mille portanti con una manciata di colpi di clock, senza utilizzare complessi filtri di canale.

Ogni singola portante viene modulata utilizzando uno schema di modulazione digitale come il QPSK (che coincide con il 4-QAM) o 16-QAM.

Un'altra tecnica utilizzata per minimizzare i disturbi sul canale è quella di introdurre un intervallo di guardia in coda a ogni simbolo. La durata del simbolo viene aumentata di un certo intervallo in cui la portante modulata viene ripetuta. Se una replica ritardata del segnale dovesse arrivare al ricevitore (per effetto dei cammini multipli), questa sarebbe completamente annullata se il ritardo fosse inferiore all'intervallo addizionale di guardia introdotto. Naturalmente, un maggiore intervallo di guardia permette di proteggere il segnale in condizioni peggiori, ma sacrifica parzialmente la banda "sprecando" del tempo a trasmettere informazioni ridondanti. Al contrario, con un intervallo di guardia molto piccolo, si aumenta il bitrate trasmesso, ma si protegge meno dai disturbi dovuti ai cammini multipli.

Per terminare questa breve panoramica sulle caratteristiche dell'OFDM, citiamo uno svantaggio derivante dall'utilizzo di questo tipo di modulazione. Se si guarda un segnale OFDM nel tempo, formato dalla somma delle molteplici portanti, ognuna modulata variando la sua fase e/o ampiezza, si vedrebbe un qualcosa di mediamente molto simile al rumore, con molti picchi. Il rapporto tra picco del segnale e suo valor medio è molto alto rispetto ad altri tipi di modulazione. La motivazione di ciò è che, a seconda dello stream binario in trasmissione, molte portanti possono tra loro sommarsi in modo costruttivo dando vita a un segnale molto alto.

Oltre a essere una nota relativamente interessante, è di capitale importanza per chi progetta un trasmettitore di potenza che deve utilizzare necessariamente degli stadi di amplificazione che introducono delle non linearità. Queste non linearità vanno a distorcere soprattutto questi picchi che, tutto sommato, fanno parte del segnale, rendendo più difficoltosa la ricezione. Un modo per risolvere questo problema è quello di utilizzare amplificatori RF maggiormente lineari, far lavorare i dispositivi a un livello di potenza inferiore o "spuntare" i picchi del segnale in modo controllato.

OFDM nel DVB-T

Lo standard DVB-T prevede un numero di portanti pari a 1705 (modalità 2K) oppure 6817 (modalità 8K). Le portanti possono essere modulate con uno schema QPSK, 16-QAM o 64-QAM. L'intervallo di guardia può essere scelto tra 1/4, 1/8, 1/16 e 1/32 della durata del simbolo. Inoltre, prima della modulazione, il modulatore manipola i bit del flusso in ingresso aggiungendo un codice convoluzionale (code rate) che permette al ricevitore di correggere automaticamente alcuni errori. È chiaro che il consorzio DVB ha voluto creare uno standard altamente configurabile e personalizzabile da parte del broadcaster che potrà sce-

Tabella 1 - bitrate (in Mbit/s) netto massimo per ogni possibile configurazione (larghezza di banda 8 MHz)

MODULAZIONE	CODE RATE	INTERVALLO DI GUARDIA 1/4	1/8	1/16	1/32
QPSK	1/2	4,98	5,53	5,85	6,03
	2/3	6,64	7,37	7,81	8,04
	3/4	7,46	8,29	8,78	9,05
	5/6	8,29	9,22	9,76	10,05
	7/8	8,71	9,68	10,25	10,56
16-QAM	1/2	9,95	11,06	11,71	12,06
	2/3	13,27	14,75	15,61	16,09
	3/4	14,93	16,59	17,56	18,10
	5/6	16,59	18,43	19,52	20,11
	7/8	17,42	19,35	20,49	21,11
64-QAM	1/2	14,93	16,59	17,56	18,10
	2/3	19,91	22,12	23,42	24,13
	3/4	22,39	24,88	26,35	27,14
	5/6	24,88	27,65	29,27	30,16
	7/8	26,13	29,03	30,74	31,67

gliere, di volta in volta, il compromesso migliore tra bitrate netto trasmesso e protezione dai disturbi (quindi distanza raggiungibile). Oltre ai parametri già definiti, è possibile scegliere la larghezza di banda del canale RF tra 5, 6, 7 e 8 MHz. La **tabella 1**, presente anche nel documento ufficiale dello standard (EN 300 744), indica il bitrate netto massimo che può essere utilizzato per ogni configurazione (considerando una larghezza di banda di 8 MHz). Come si può vedere, il massimo bitrate utilizzabile è di 31,67 Mbit/s (circa 5 programmi codificati con un bitrate di 6 Mbit/s ognuno).

Il modulatore DVB-T

Un trasmettitore DVB-T è un apparato elettronico che accetta in ingresso un *transport stream* (secondo le specifiche viste sopra) e produce in uscita un segnale RF da inviare in antenna. Naturalmente la potenza del segnale d'uscita potrà essere scelta in base alla copertura voluta e alle autorizzazioni da parte degli enti preposti (in Italia, il Ministero delle Poste e Telecomunicazioni), considerando che una potenza maggiore significa un sensibile aumento di costo dell'amplificatore RF e dei consumi energetici.

Normalmente un modulatore DVB-T lavora in due fasi: a partire dalla banda base genera un segnale d'uscita alla frequenza intermedia (IF, *Intermediate Frequency*) tipicamente di 70 MHz. Un oscillatore locale permette successivamente lo spostamento da IF a RF (naturalmente l'oscillatore locale deve possedere elevate caratteristiche di "pulizia" della portante per non "rovinare" il segnale OFDM). Il già citato standard (EN 300 744) prevede una serie di elaborazioni in cascata del *transport stream*, dall'ingresso verso l'uscita.

Una primissima elaborazione consiste nel rendere maggiormente casuale la sequenza di bit in ingresso per avere un sufficiente numero di transizioni 1/0 e viceversa (*energy dispersal*). A tale scopo si utilizza un metodo matematico che si fonda sulle sequenze pseudo-casuali denominate PRBS (*Pseudo-Random Bit Sequence*). Ovviamente il processo è perfettamente invertibile dal ricevitore (è per questo che le sequenze sembrano casuali, ma non lo sono, sono appunto pseudo-casuali).

Fino a questo punto la pacchettizzazione da 188 byte, con in testa un byte di sincronismo, non è stata modificata. Il passo successivo è quello di aggiungere (*outer encoder*) un codice d'errore (un parente del bit di parità) di 16 byte a ogni pacchetto di 188 byte, ottenendo pacchetti di 204 byte. Questo codice d'errore è di tipo *Reed-Solomon* (abbreviato con RS) e ha la pregevole caratteristica di permettere al ricevitore di correggere automaticamente fino a 8 byte errati in un pacchetto, indipendentemente dalla posizione dei bit errati (quindi anche se avvengono a burst, cioè tutti insieme). Successivamente avviene un mescolamento (*outer interleaving*) dei byte per renderli ancora più casuali ed evitare che un errore prolungato nel tempo colpisca una sequenza lunga di byte consecutivi. Quindi vengono aggiunti altri codici d'errore (inner encoder) di tipo convoluzionale, spesso indicato come FEC. Il rapporto tra i bit utili e i bit di correzione aggiunti può essere, come già visto, 1/2, 2/3, 3/4, 5/6 o 7/8. Ancora una volta il flusso di bit viene mescolato (*inner interleaving*) sempre nell'ottica di recuperare gli eventuali errori in ricezione.

Successivamente la sequenza dei bit viene "mappata" (*mapper*) nei corrispondenti simboli dello schema di modulazione digitale scelto (QPSK, 16-QAM, 64-QAM). Giusto come esempio, in **figura 4** si riporta la costellazione 16-QAM, utilizzata nel DVB-T, con la relativa codifica dei bit per ogni punto. Questa codifica è di tipo Gray, in quanto, per ogni simbolo, la codifica di quelli vicini differisce soltanto di un bit. In questo modo, se un disturbo sposta il punto originale in uno limitrofo, si ha al massimo un bit di errore (ovviamente, se il disturbo è notevole, il punto si può spostare parecchio nella costellazione e più bit possono essere ricevuti in modo errato).

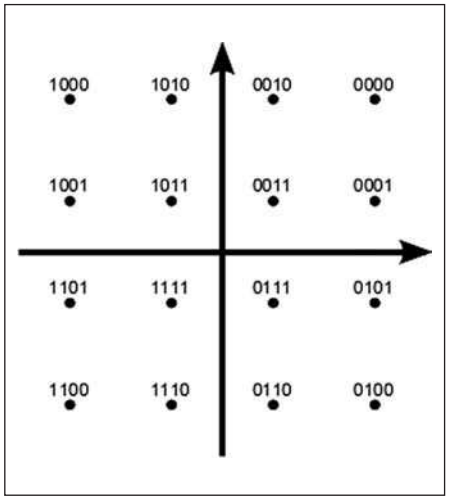


Figura 4: costellazione 16-QAM con la relativa codifica Gray dei bit.

Infine vengono raggruppati i simboli in frame da 68 simboli (*frame adaptation*) e modulati in OFDM (*OFDM modulator*). L'uscita del modulatore a frequenza intermedia IF viene convertita in canale RF (*up-converter*), eventualmente amplificata (*high power amplifier*) e inviata al sistema d'antenna.
Ma come potrebbe apparire questo segnale DVB-T su un analizzatore di spettro? In **figura 5** è mostrato lo spettro teorico (calcolato matematicamente) di un segnale DVB-T con larghezza di banda 8 MHz e modalità 2k e 8k. Come si può osservare, il canale RF a disposizione è utilizzato in modo molto efficiente e il segnale, in banda, sembra essere rumore bianco.
Per poter convivere con altri segnali DVB-T o analogici, soprattutto se presenti nei canali adiacenti, la normativa prevede dei valori massimi dei segnali ai limiti e all'esterno del canale. Se il trasmettitore non dovesse rientrare in questi limiti (cosa che succede praticamente sempre per colpa dell'amplificatore non lineare), sarà necessario utilizzare un apposito filtro d'uscita.

Il set-top-box

Come tutti sanno, durante la prima fase di diffusione della televisione digitale, è necessario dotarsi di un apposito apparecchio, chiamato set-top-box (STB), per poter ricevere il segnale digitale e trasformarlo in uno compatibile con le televisioni analogiche. Questa prima fase è ormai terminata, visto che in commercio sono già disponibili, da parecchio tempo, televisioni che possono ricevere direttamente il segnale digitale (in pratica, hanno un piccolo set-top-box integrato).
Il funzionamento del ricevitore DVB-T è perfettamente speculare a quello del modulatore, in modo da ottenere nuovamente il TS originale in ingresso al modulatore. Naturalmente il ricevitore si servirà di tutti i codici per la correzione degli errori inseriti nel modulatore per ripristinare i bit ricevuti in modo errato. Normalmente la sezione di ricevitore che accetta in ingresso il segnale RF e genera in uscita il TS MPEG-2 è fatta da un unico modulo, denominato *frontend*, al cui interno sono presenti un *tuner*, che trasforma il segnale RF in ingresso in un segnale a frequenza intermedia IF, e un demodulatore che, a partire dal segnale IF, genera il *transport stream* in uscita. Entrambi i chip vengono normalmente controllati dal processore principale mediante bus I_2C. Uno dei maggiori produttori attivi in questo campo è NXP (ex Philips Semiconductor).
A partire dal TS, i set-top-box DVB-S (satellitari) e DVB-T sono identici, proprio grazie alla infrastruttura di base simile. In genere si ha una soluzione single-chip altamente integrata per minimizzare il costo del prodotto rivolto al consumatore finale. Questo chip contiene un micro-

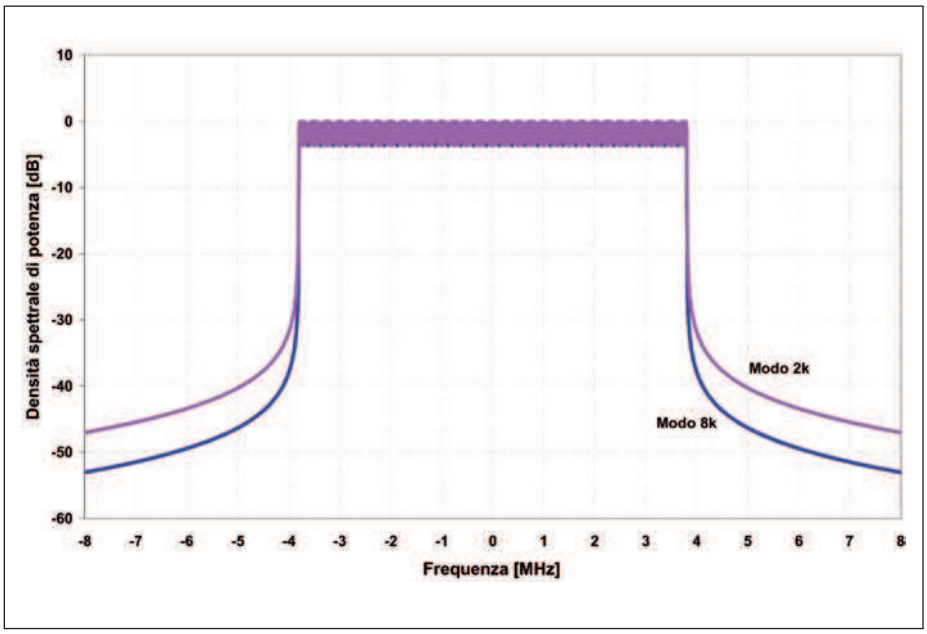


Figura 5: spettro di un segnale DVB-T con larghezza di banda 8MHz e modalità 2k e 8k.

processore, che si preoccupa del controllo, e uno o due DSP per l'elaborazione del segnale audio/video. Particolarmente attivo in questo settore è ST, con tutta una serie di processori presenti in molti set-top-box commerciali. Dal TS vengono estratti i flussi audio e video inerenti al programma scelto dall'utente (si ricordi che un TS può contenere un bouquet di più programmi) e inviati ai rispettivi decoder. A questo punto il segnale video può essere convertito in analogico su connettore video composito (per esempio, su connettore SCART o RCA) o S-video (su connettore Mini-DIN). Allo stesso modo, il segnale audio può essere convertito in analogico su connettori RCA (left/right) oppure su SCART. In alcuni set-top-box i segnali possono rimanere digitali ed essere trasformati nel nuovo standard di interconnessione multimediale chiamato HDMI.

COSA CI RISERVA IL FUTURO

Alta definizione

L'infrastruttura digitale del DVB-T permette di utilizzare altre tecniche per ottimizzare la banda richiesta da un programma audio/video. Per esempio, è possibile utilizzare nuove tecniche di compressione, come MPEG-4 (detta anche H.264), che aumentano il fattore di riduzione e permettono di convogliare più programmi a parità di bitrate, oppure programmi a maggiore risoluzione video a parità di bitrate (alta definizione). Una tipica risoluzione ad alta definizione è 1280x720 oppure 1920x1080. Il maggior numero di pixel, e quindi di informazioni da trasmettere, viene compensato dal maggior fattore di compressione dell'MPEG-4.

DVB-T2

Il consorzio DVB, continuando sulla sua strada di proliferazione di standard, ha creato già la versione 2 della televisione digitale, denominata DVB-T2, che potrà coesistere con l'attuale versione DVB-T durante il suo sviluppo. Non ci sono grandi modifiche rispetto al primo standard, infatti, si continua a usare una modulazione OFDM. Le novità riguardano soprattutto una maggiore efficienza dei codici di errore che passano da codici convoluzionale e *Reed-Solomon*, in codici LDPC (*Low Density Parity Check*) e BCH (*Bose-Chaudhuri-Hocquengham*), usati anche nel DVB-S2. L'altra novità riguarda una maggiore flessibilità di configurazione dei parametri. Per esempio, è stata aggiunta la modulazione 256-QAM; gli intervalli di guardia 19/256, 19/128 e 1/128; le modalità 1k, 4k, 16k, 32k (con un numero diverso di portanti). Tutto ciò, insieme ad altre piccole differenze, promette una trasmissione ancora più robusta anche in presenza di rumori più forti.

PER *approfondire...*

La documentazione principale di riferimento è sicuramente l'insieme degli standard prodotti dal consorzio DVB. Nel sito ufficiale (www.dvb.org) esiste una sezione apposita che elenca tutte le specifiche, normalmente pubblicate dall'ente europeo ETSI (European Telecommunications Standards Institute). Tutti i documenti sono liberamente scaricabili direttamente dal sito www.etsi.org, ovviamente in inglese. Purtroppo il consorzio DVB è noto per essere molto prolifico ed è quindi molto difficile districarsi tra i mille documenti prodotti. È possibile iniziare leggendo un documento scritto proprio dal consorzio per aiutare i novizi a orientarsi: TR 101 200: A *guideline for the use of DVB specifications and standards*. Qui di seguito un brevissimo elenco degli standard principali.

- ETR 154: linee guide sull'utilizzo dello standard MPEG-2 nelle applicazioni DVB;

- EN 300 468: descrizione delle informazioni di servizio (DVB-SI, Service Informations) che sono aggiunte nel transport stream MPEG per permettere all'utente di conoscere i servizi presenti nel flusso di bit ricevuto (principalmente i programmi presenti nel bouquet, ma anche il nome delle stazioni, gli eventi in onda e altro ancora);

- EN 300 744: è lo standard principale per chi vuole approfondire le tematiche riguardanti esclusivamente la trasmissione terrestre del transport stream (rappresenta il documento principale del DVB-T);

Per chi non conoscesse l'inglese, oltre al materiale introduttivo presente sulle pagine di Wikipedia, si consigliano vivamente i numerosi articoli sull'argomento pubblicati dalla rivista gratuita del Centro Ricerche RAI, intitolata Elettronica e Telecomunicazioni, in particolare:

- V. Mignone, A. Morello, M. Visintin, Lo standard DVB-T per la televisione digitale terrestre; aprile 2002

- V. Mignone, La modulazione COFDM; dicembre 2005

CONCLUSIONI

Chi ha raggiunto questo punto dell'articolo avrà sicuramente capito che l'argomento è piuttosto vario e difficilmente riassumibile in poche pagine e purtroppo sono stati fatti parecchi tagli sugli argomenti meno importanti. Citiamo come esempio una serie di argomenti non trattati:

- l'accesso condizionato, mediante il quale è possibile cifrare un programma in modo che solo utenti selezionati (cioè paganti) possano decifrare e vedere;
- l'interattività, che fornisce all'utente la possibilità di interagire con la televisione, per esempio rispondendo con il suo telecomando a un quiz in onda, oppure eseguendo applicativi appositi direttamente sul proprio televisore;
- la navigazione internet, resa possibile aggiungendo i pacchetti IP all'interno del *transport stream* MPEG2;
- la televisione mobile, che permette di ricevere un programma televisivo anche su dispositivi palmari, telefoni cellulari, notebook, anche in movimento (per esempio, all'interno di un treno);
- SFN (*Single Frequency Network*), cioè la possibilità di poter usare la stessa frequenza anche per trasmettitori adiacenti e quindi anche per tutta la rete, senza che questi creino disturbi (cosa praticamente impossibile con la televisione analogica). ❑

CODICE MIP 2757071

di GIOVANNI DI MARIA

CINQUE MELODIE CON UN PIC

Questo mese realizziamo un progetto sonoro nel quale utilizziamo un microcontrollore PIC abbinato al nuovo compilatore MikroBasic Pro della MikroElektronika per riprodurre ben cinque melodie musicali

Lo abbiamo detto più volte, un microcontrollore, per il fatto che può generare tensioni in uscita, è capace di produrre note musicali. La rapida cadenza di accensioni e spegnimenti di una porta produce una frequenza che può essere udibile dall'uomo. L'omogenea e coordinata gestione delle frequenze prodotte (ed eventuali pause) dà luogo alla musica. In precedenti puntate abbiamo visto come generare suoni, beep, toni telefonici, segnali di avvertimento e semplici musichette. Questa volta studiamo il metodo per riprodurre ben cinque brani musicali a comando, con diverse velocità di esecuzione.

IL GADGET

L'utilizzo del prototipo che ci accingiamo a illustrare può essere vario. Si può considerare un gadget a tutti gli effetti, in quanto l'aspetto ludico e il fine divertente non può che confermarne questa funzionalità. Ma è soprattutto l'aspetto didattico quello che più ci preme, in quanto esso ci coinvolge per l'apprendimento di tanti concetti che si possono così riassumere:

- come riuscire a memorizzare cinque musichette all'interno del PIC;
- come riuscire a trasformare una canzone in un insieme di dati numerici;
- come riuscire a tradurre uno spartito in una successione di dati numerici;
- e soprattutto come effettuare tali operazioni con un microcontrollore.

PANORAMICA GENERALE

Innanzitutto esaminiamo da un punto di vista generale il progetto, evidenziando quelle che sono le componenti software, hardware e di comportamento funzionale. Questo esame ci servirà per comprendere a fondo il funzionamento del sistema e affrontare con maggiore cognizione le successive fasi realizzative.

L'hardware

La parte elettronica è formata naturalmente da un microcontrollore che sovrintende a tutte le gestioni logiche delle procedure. Esso contiene fisicamente anche le note musicali, in quanto la sua memoria è abbastanza capiente. Coordina altresì le operazioni di avvio di una canzone, scelta dall'operatore, in accordo con la velocità gradita. Inoltre il sistema è dotato di alcuni pulsanti che hanno la funzione di attivare la musica prescelta. In più, due diodi LED danno un'indicazione precisa sullo stato di esecuzione ovvero di stand-by del prototipo.

Il software

Il programma, come si vedrà nelle successive sezioni, è semplice. Esso gestisce i cinque brani musicali in modo da attivarli a piacimento.

I BRANI MUSICALI SCELTI

Per la realizzazione del progetto sono state scelte le seguenti canzoni:

- *Fra Martino;*
- *Jingle Bells;*
- *Tu scendi dalle stelle;*
- *Tanti auguri a te;*
- *Inno di Mameli.*

Si tratta, come si vede, di semplici brani conosciuti da tutti. Naturalmente si può adottare qualsiasi altra canzone, ma ciò comporta la ricodifica del programma sorgente nonché di tutte le note musicali a essa appartenenti.

Qualità di riproduzione

Ci teniamo a precisarlo subito: non stiamo realizzando un sistema di riproduzione musicale di tipo Wave e neanche digitalizzato. Stiamo semplicemente riproducendo delle frequenze cadenzate costituite, a basso

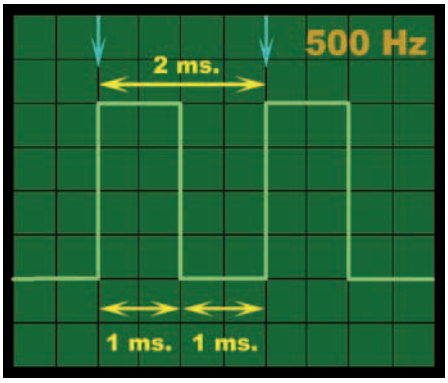


Figura 1: onda quadra di 500 Hz.

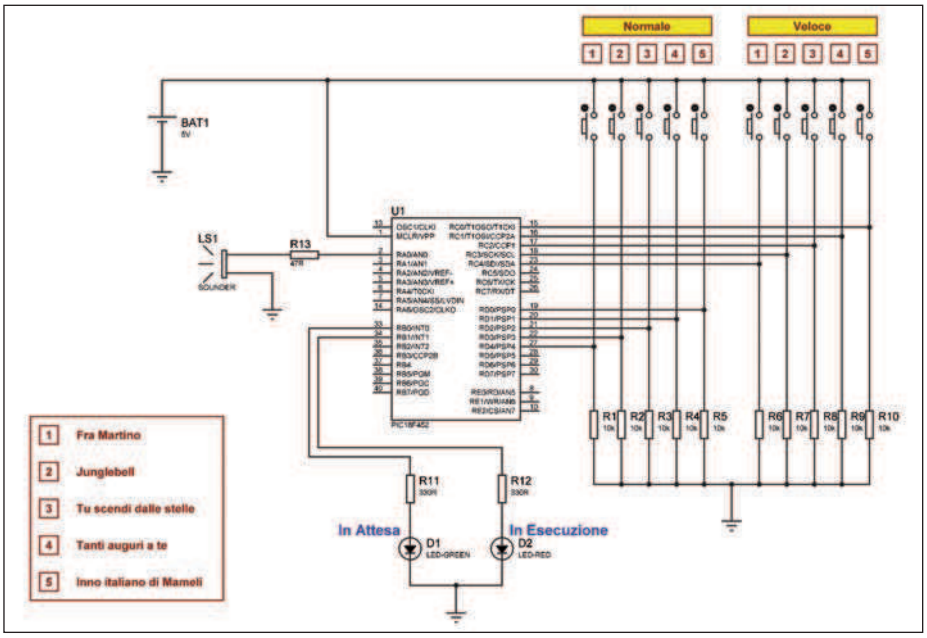


Figura 2: schema elettrico del generatore di cinque melodie.

Tabella 1 - Nomi e frequenze delle note (in hertz)

	OTTAVE					
NOTA	2	3	4	5	6	7
Do	65,4	130,8	261,6	523,3	1046,5	2093
Do#	69,3	138,6	277,2	554,4	1108,7	2217,5
Re	73,4	146,8	293,7	587,3	1174,7	2349,3
Re#	77,8	155,6	311,1	622,3	1244,5	2489
Mi	82,4	164,8	329,6	659,3	1318,5	2637
Fa	87,3	174,6	349,2	698,5	1396,9	2793,8
Fa#	92,5	185	370	740	1480	2960
Sol	98	196	392	784	1568	3136
Sol#	103,8	207,7	415,3	830,6	1661,2	3322,4
La	110	220	440	880	1760	3520
La#	116,5	233,1	466,2	932,3	1864,7	3729,3
Si	123,5	246,9	493,9	987,8	1975,5	3951,1

livello, da stati logici alti e bassi. Pertanto la musica prodotta è caratterizzata da un timbro tipico del suono a onda quadra e monofonica. Viene riprodotta cioè una nota per volta. Ciò comunque non toglie nulla al risultato molto preciso e intonato, simile a quello riprodotto dalle classiche luci utilizzate per allestire un albero di Natale.

Caratteristiche della riproduzione musicale del circuito

- Tipo di onda musicale: onda quadra;
- duty cycle: 50%;
- tipo di suono: monofonico;
- intonazione: eccellente.

SUONO, NOTA E MUSICA

Il suono è una vibrazione meccanica che, propagandosi in un mezzo elastico, ad esempio l'aria, provoca una percezione da parte del nostro sistema uditivo. La frequenza deve essere compresa tra 20 Hz e 20.000 Hz per poter essere percepita dall'uomo. La frequenza cataloga il suono in sette note (DO, RE, MI, FA, SOL, LA, SI) distribuite tra diverse ottave (vedi **tabella 1**).

LA FUNZIONE DEL SISTEMA

Esaminiamo a fondo la funzionalità del sistema e delle relative interazioni con l'utente. Esso, come abbiamo detto in precedenza, è dotato di ben dieci pulsanti normalmente aperti. Le canzoni sono cinque e possono essere riprodotte in due diverse velocità (normale e veloce). Pertanto ogni canzone può essere attivata da due pulsanti differenti, uno per la riproduzione normale e l'altro per quella veloce. All'accensione, il circuito non riproduce alcun brano musicale e questo fatto è sottolineato dall'accensione del diodo LED verde, che ne indica lo stato di attesa. Quando si preme un tasto per la riproduzione, la musica viene eseguita sino al termine della stessa e l'evento è evidenziato dall'accensione del diodo LED rosso. In tal caso, nessun altro motivo musicale può essere selezionato, fino al completo esaurirsi del precedente.

LISTA COMPONENTI

Sigla	Valore
Bat	Batteria 5 volt
Ls1	Piccolo altoparlante 8 ohm
R1÷R10	10 kohm
R11÷R12	330 ohm
R13	47 ohm
D1	Diodo LED VERDE
D2	Diodo LED ROSSO
U1	PIC 18F452
	10 pulsanti n/a
	2 condensatori 22 pF
	1 quarzo 4 MHz

SCHEMA ELETTRICO

Lo schema elettrico raffigurato è di principio e può essere adattato praticamente a qualsiasi altro tipo di microcontrollore. L'unica condizione richiesta è quella di utilizzare una MCU capiente, nella quale possano allocarsi correttamente tutti i dati relativi alle cinque canzoni. Le dieci porte che sono collegate ai relativi pulsanti sono ingressi collegati ad altrettante resistenze di pull-down che assicurano un potenziale nullo in caso di mancata pressione. Infine i diodi LED di segnalazione di stato (verde e rosso) sono collegati alle rispettive resistenze di limitazione di corrente.
Anche se lo schema elettrico non riporta il quarzo e i condensatori di oscillazione, occorre prevederli secondo gli utilizzi standard dei microcontrollori. Occorre prevedere altresì la corretta alimentazione TTL alla MCU. Il microcontrollore utilizzato è il PIC 18F452, scelto per la sua

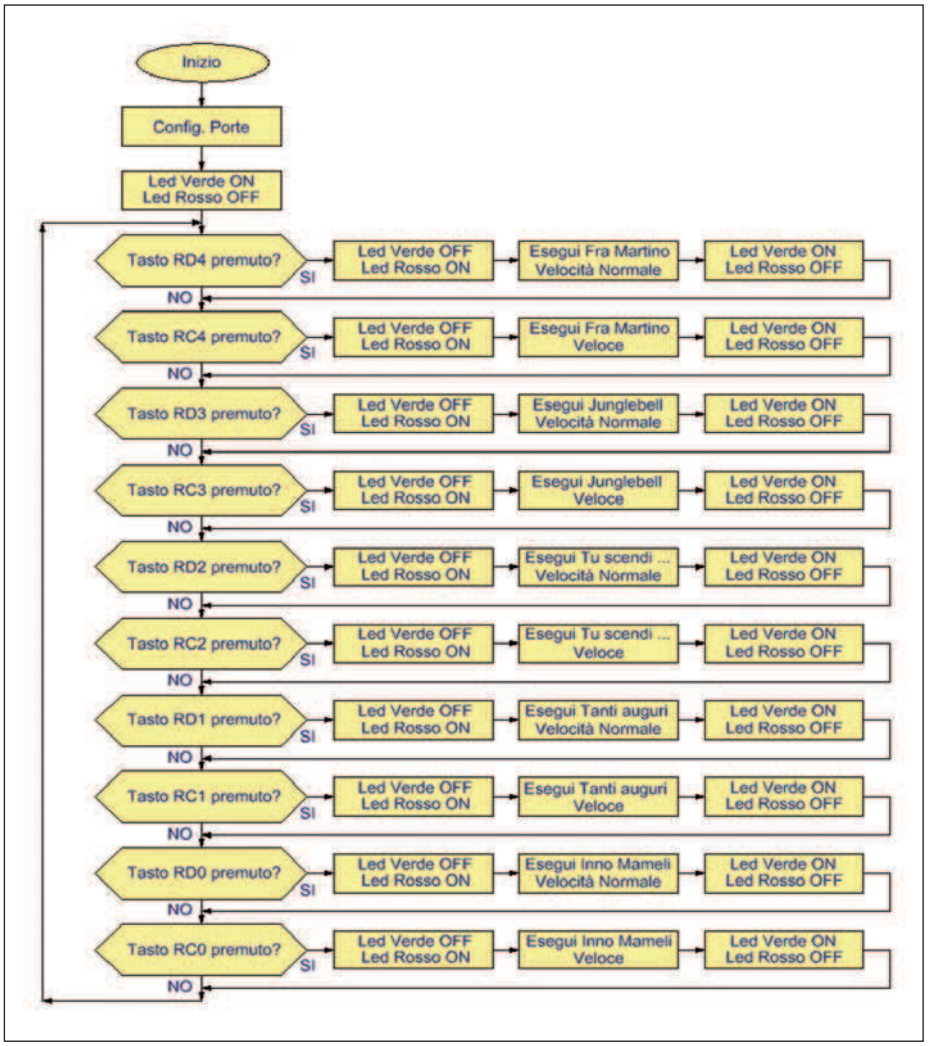


Figura 3: flow chart dell'intera procedura.

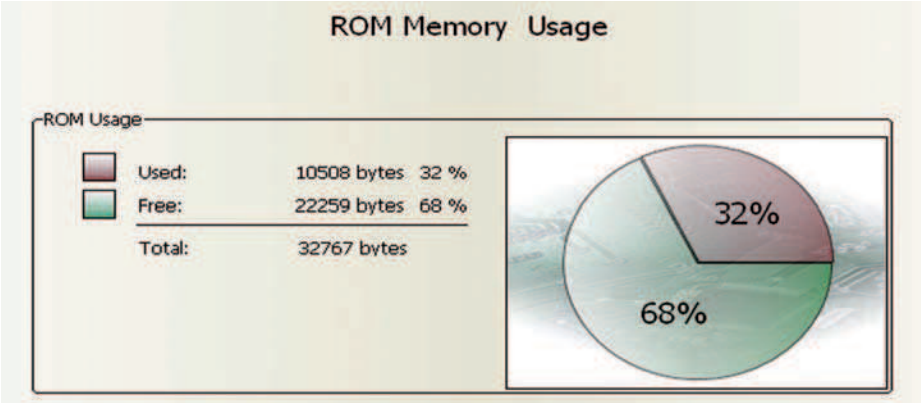


Figura 4: analisi statistica dell'occupazione della memoria.

grande capienza di memoria.

IL PROGRAMMA

Il listato è molto lungo per via delle informazioni musicali memorizzate. Per questo motivo è stato scelto un microcontrollore molto capiente. Il programma è strutturato in diverse procedure, richiamabili dalla funzione principale. Tali procedure sono adibite all'esecuzione dei brani musicali e per questo motivo sono state intestate con il titolo della relativa canzone. Altre due procedure hanno il compito di impostare lo stato dei diodi LED, rispettivamente in modalità di attesa oppure di esecuzione in corso. L'intero processo itera sul controllo continuo dei pulsanti che determinano la canzone da suonare e la relativa velocità di esecuzione.

L'intero programma è scritto per il nuovo compilatore Mikrobasic Pro della Mikroelektronika e, nella fattispecie, è stata utilizzata la versione 2.50.

Come detto prima, l'utilizzo delle procedure (simile alle funzioni ma che non ritornano un valore al programma chiamante) permette la modularità dell'intero sistema e un'agevole manutenzione. Segue un elenco delle procedure utilizzate, con relativa funzionalità:

- procedura *framartino*: riproduce *Fra Martino*;
- procedura *jinglebell*: riproduce *Jingle Bells*;
- procedura *tuscendidallestelle*: riproduce *Tu scendi dalle stelle*;
- procedura *tantiauguri*: riproduce *Tanti auguri a te*;
- procedura *innoitalia*: riproduce l'*Inno di Mameli*;
- procedura *leds_durante_attesa*: spegne LED rosso e accende LED verde;
- procedura *leds_durante_esecuzione*: spegne LED verde e accende LED rosso.

Il **listato 1** è abbastanza lungo, circa 450 linee di programma ed è possibile scaricarlo dal sito di Fare Elettronica. La maggior parte di codice è adibita alla codifica delle note musicali delle cinque canzoni. Durante la compilazione, l'ambiente di sviluppo produrrà alcuni *warning* dovuti a un adattamento tra due diverse tipologie di dato (*typecast*). In particolare, ciò avviene all'interno delle procedure, laddove sono previste delle moltiplicazioni o divisioni con un numero decimale. Si potrebbero evitare implementando un adattamento di *cast*. Si consulti il manuale del Mikrobasic Pro a tale proposito.La compilazione produce un eseguibile abbastanza corposo che occupa molta memoria del micro. Le relative immagini statistiche mostrano l'occupazione dell'intero codice nonché delle singole procedure e funzioni.

LE FUNZIONI MUSICALI BUILT-IN

Premessa

Il compilatore Mikrobasic Pro è dotato di alcune funzioni "preimpostate" che generano, con pochissime righe di program-

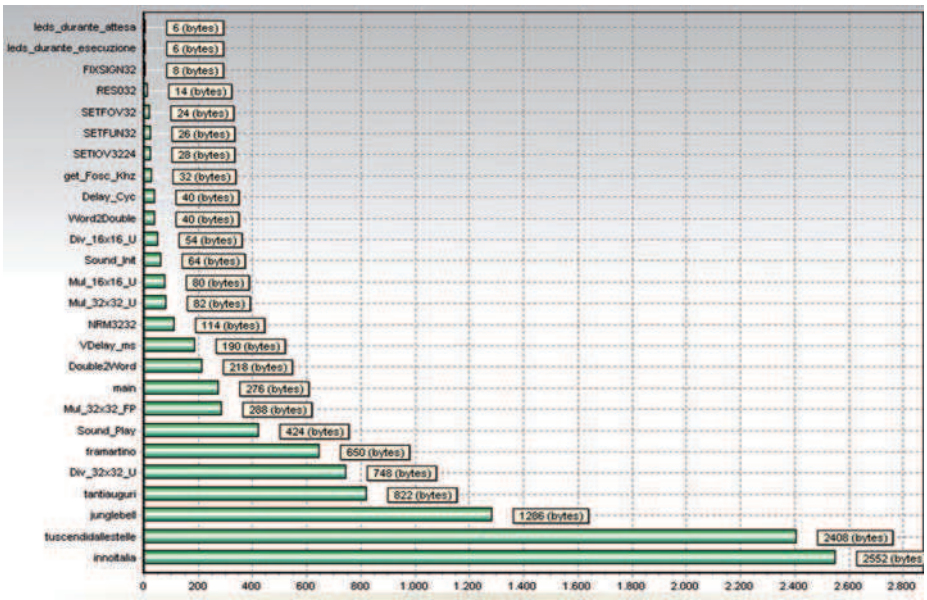


Figura 5: analisi statistica dell'occupazione delle singole procedure e funzioni del programma.

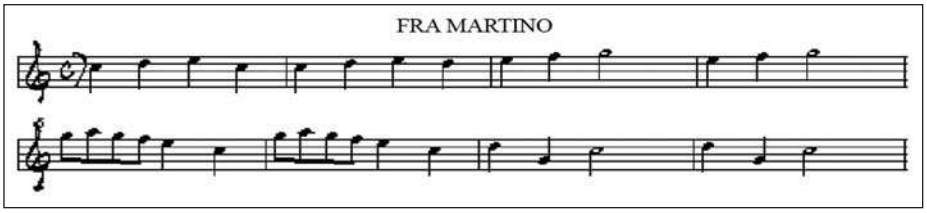


Figura 6: lo spartito di Fra Martino.

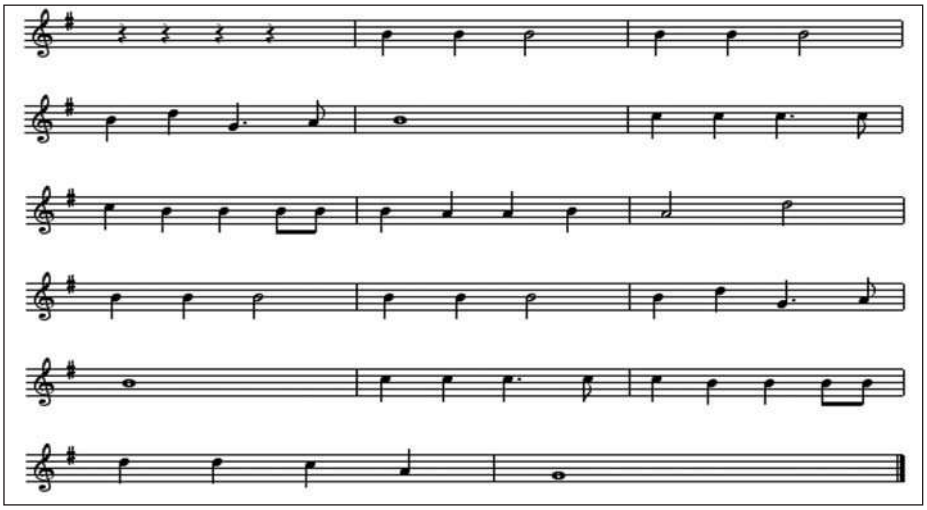


Figura 7: lo spartito di Jingle Bells.

mazione, qualsiasi frequenza. Con le vecchie versioni del compilatore, tali funzioni erano complicate da utilizzare, poiché i parametri relativi al suono non corrispondevano formalmente alla frequenza e alla durata delle note, bensì si trattava di valori numerici da calcolare. Allo scopo si veda la puntata del corso di basic per PIC relativa alla generazione sonora. Con la versione Pro dell'ambiente di sviluppo, generare una nota è molto più semplice e immediato.

Le funzioni

Le funzioni supportate dal Mikrobasic Pro, almeno nella versione attuale, sono due:

- Sound_init*(porta,pin)*;
- Sound_play*(frequenza,durata)*.

La prima predispone il microcontrollore ad accettare su una porta (ad es. PORTA o PORTB) un dispositivo sonoro di output (altoparlante o buzzer). Tale dispositivo deve essere collegato sul pin specificato dal secondo parametro. Per esempio, la funzione:

```
Sound_init(PORTB,0)
```

prepara la PORTB, e precisamente il pin RB0, alla produzione di suoni.
La seconda invece si occupa della vera e propria riproduzione della nota, generando un'onda quadra di opportuna frequenza. Vediamo come si usa. Come si vede dalla sintassi della funzione, essa accetta (tra parentesi) due parametri: il primo indica la frequenza da generare (espressa in Hz), il secondo indica invece la durata della stessa (espressa in millisecondi). Per esempio, la funzione:

```
Sound_play(1000,500)
```

serve per generare una nota 1000 Hz per mezzo secondo.

I CINQUE MOTIVI MUSICALI

I brani sono stati scelti tra quelli conosciuti universalmente e sono semplici da cantare e riprodurre.

Come tradurre lo spartito in dati digitali

Per preparare l'insieme dei dati digitali, provenienti dallo spartito, è opportuno essere in possesso di una minima conoscenza di notazione musicale e divisione. Solo in questo modo, infatti, è possibile conoscere il nome della nota (con la relativa frequenza) e, soprattutto, la sua durata. Passiamo pertanto in rassegna le fasi necessarie alla traduzione di uno spartito in un insieme di parametri digitale, da inserire poi all'interno del programma musicale.

1° passo

Occorre stabilire la durata di una figura musicale. Questo dato è solitamente espresso all'inizio dello spartito e determina la durata di una figura prestabilita. Formalmente esso viene indicato con la notazione visibile in **figura 11**. L'esempio chiarisce che in un minuto vi sono 80 semiminime, la cui durata equivale a 60/80 = 0,75 sec = 750 ms.

2° passo

Stabilita la durata della semiminima, è possibile risalire a tutte le altre figure di durata, secondo la notazione musicale. Occorre in altri termini effettuare la divisione musicale.

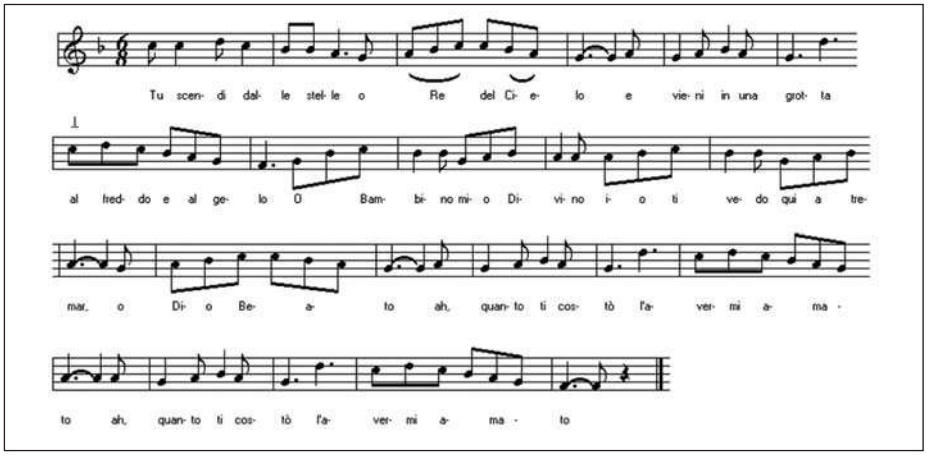


Figura 8: lo spartito di Tu scendi dalle stelle.

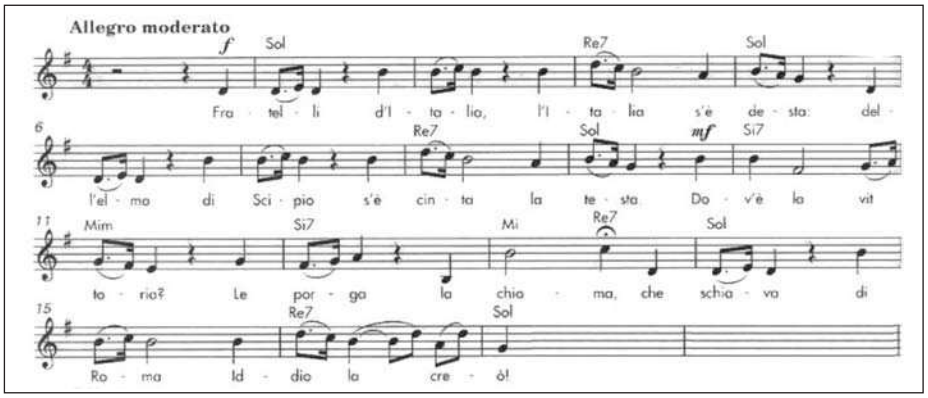


Figura 10: lo spartito dell'Inno di Mameli.

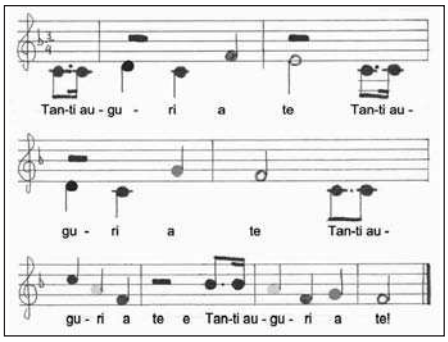


Figura 9: lo spartito di Tanti auguri a te.

Si consulti a tale proposito Internet oppure, in alternativa, si può acquistare uno dei tanti metodi musicali in commercio.

3° passo

Adesso si può scrivere la procedura, sapendo che nel nostro esempio il parametro passato tra parentesi indica proprio la durata di una semiminima. Per eseguire una nota, pertanto, si deve specificare il tempo di durata della semiminima, moltiplicato o diviso per un fattore di correzione, per ricavare tutte le altre figure. Il tempo deve essere espresso in millisecondi.

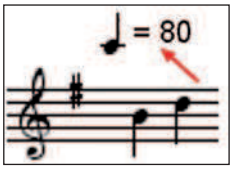


Figura 11: velocità del brano indicato sullo spartito.

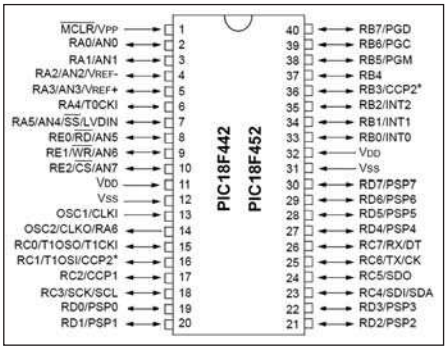


Figura 12: pinuot del PIC 18F452.

COLLAUDO

Il collaudo avviene senza problemi, dal momento che tutto il funzionamento è delegato al microcontrollore. Dopo aver dato alimentazione al circuito, è sufficiente premere uno dei dieci tasti. Essi iniziano l'esecuzione del brano con la relativa velocità scelta. Occorre notare che non è possibile avviare l'esecuzione di un brano se non è terminato il precedente; questo perché il programma è "congelato" dalle funzioni sonore per tutta la durata della musica. Lo stato di accensione dei diodi LED attesta proprio questo stato. La funzione dei tasti è descritta nello schema elettrico.

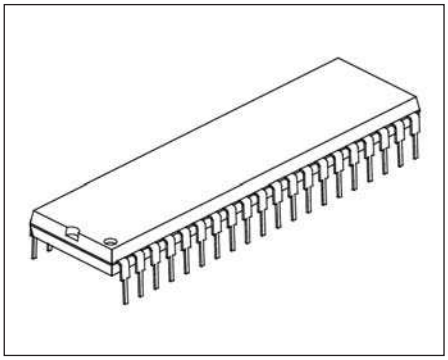


Figura 13: il contenitore del PIC 18F452.

Features	PIC18F452
Operating Frequency	DC - 40 MHz
Program Memory (Bytes)	32K
Program Memory (Instructions)	16384
Data Memory (Bytes)	1536
Data EEPROM Memory (Bytes)	256
Interrupt Sources	18
I/O Ports	Ports A, B, C, D, E
Timers	4
Capture/Compare/PWM Modules	2
Serial Communications	MSSP, Addressable USART
Parallel Communications	PSP
10-bit Analog-to-Digital Module	8 input channels
RESETS (and Delays)	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST)
Programmable Low Voltage Detect	Yes
Programmable Brown-out Reset	Yes
Instruction Set	75 Instructions
Packages	40-pin DIP 44-pin PLCC 44-pin TQFP

Figura 14: caratteristiche del PIC 18F452.

CONCLUSIONI

Come si vede, far suonare un PIC non è poi così difficile. Si tratta di trovare il modo per trasformare uno spartito musicale in un insieme di numeri. Il lettore può esercitarsi a implementare altre canzoni, ricordando però che la memoria del micro si riempie facilmente. Buon lavoro! ❑

CODICE MIP 2757055

DEVELOPMENT TOOL...

Complete AVR development solution

Speed up your prototype development with the EasyAVR6 Development System. The EasyAVR6 comes packed with examples that makes your AVR development fast and easy.

Save time & money!

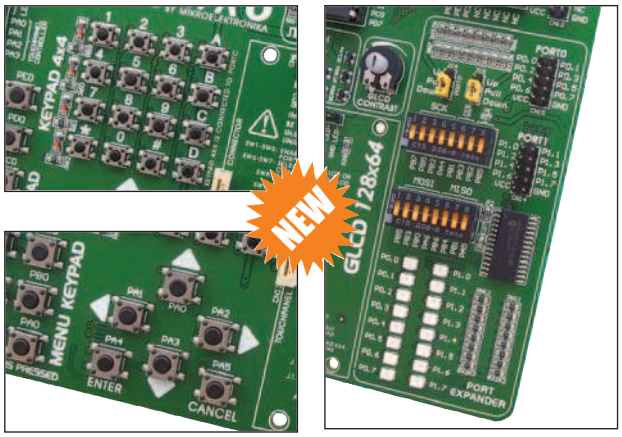
Get extra value for your money with this first-class tool for AVR development solutions. Get into the AVR world faster and easier than ever before with the EasyAVR6 Development System.

Designed to suit your needs

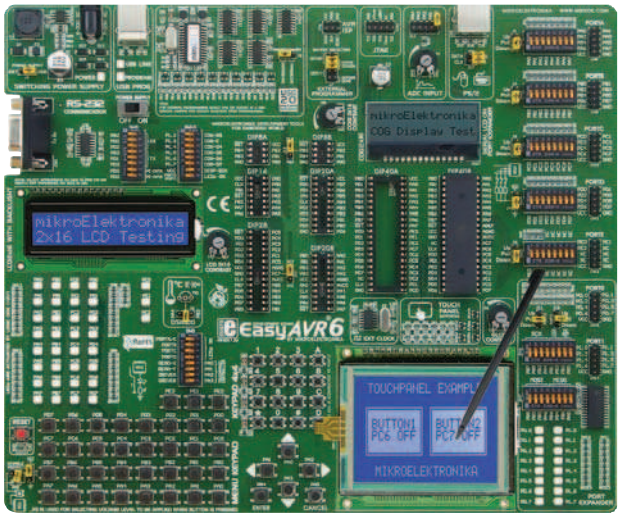
Your development time can be considerably reduced, resulting in an early prototype design and fast time-to-market for your end product.

...JUST THE WAY YOU LIKE IT !!!

Everything you've always wanted from a development tool
Experience the ease of creating your own electronic devices!



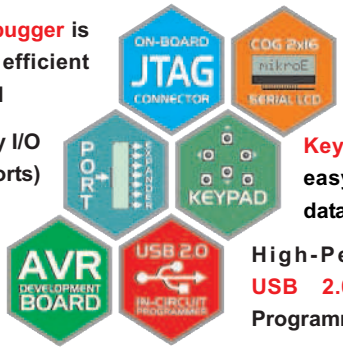
Thanks to many new features, you can start creating your own devices immediately. **EasyAVR6** supports **8-, 14-, 20-, 28- and 40- pin AVR** microcontrollers. The on-board **USB 2.0 Programmer** enables very efficient and fast AVR microcontroller programming. Examples in **C**, **BASIC** and **Pascal** are provided with the board.



Connector for the JTAG Debugger is on-board and enables efficient debugging at hardware level

Port Expander provides easy I/O expansion (2 additional ports) using serial interface

Full-featured and **user-friendly** development board for AVR microcontrollers



On-Board 2x16 Serial LCD Display

Keypad enables easy and fast data entry

High-Performance USB 2.0 On-Board Programmer

lead-free

RoHS 2002/95/EC

Find your distributor:
http://www.mikroe.com/en/distributors/

mikroElektronika
SOFTWARE AND HARDWARE SOLUTIONS FOR THE EMBEDDED WORLD

CODICE MIP 2759219

www.mikroe.com

di DANIELE CAPPA

Frequenzimetro con PIC 16F628 e display a sette SEGMENTI

Strumento "minimo" con un'ottima versatilità, è utilizzabile sia come strumento a sé stante, sia come sintonia elettronica per piccoli ricetrasmettitori autocostruiti

Le caratteristiche sono di tutto rispetto, da 1 a 60 MHz, selezione automatica delle scala (*autoranging*), cinque cifre con display a LED, possibilità di sommare o sottrarre il valore di media frequenza. Il primo frequenzimetro a PIC è stato realizzato dall'autore anni fa per utilizzarlo insieme a un vecchio TS700 Kenwood fornito di transverter in 6 metri. L'articolo completo della modifica è reperibile su www.iw1axr.eu, e lo schema utilizzato è quello pubblicato da ik3oil sul numero di maggio 2000 di Radio Rivista. In verità qualche dubbio c'era sul funzionamento di strumenti di questo tipo, negli anni si è poi imparato ad apprezzarne la versatilità, tanto che il TS700 ha ormai cambiato proprietario, ma il frequenzimetro è già stato "riutilizzato" più volte.

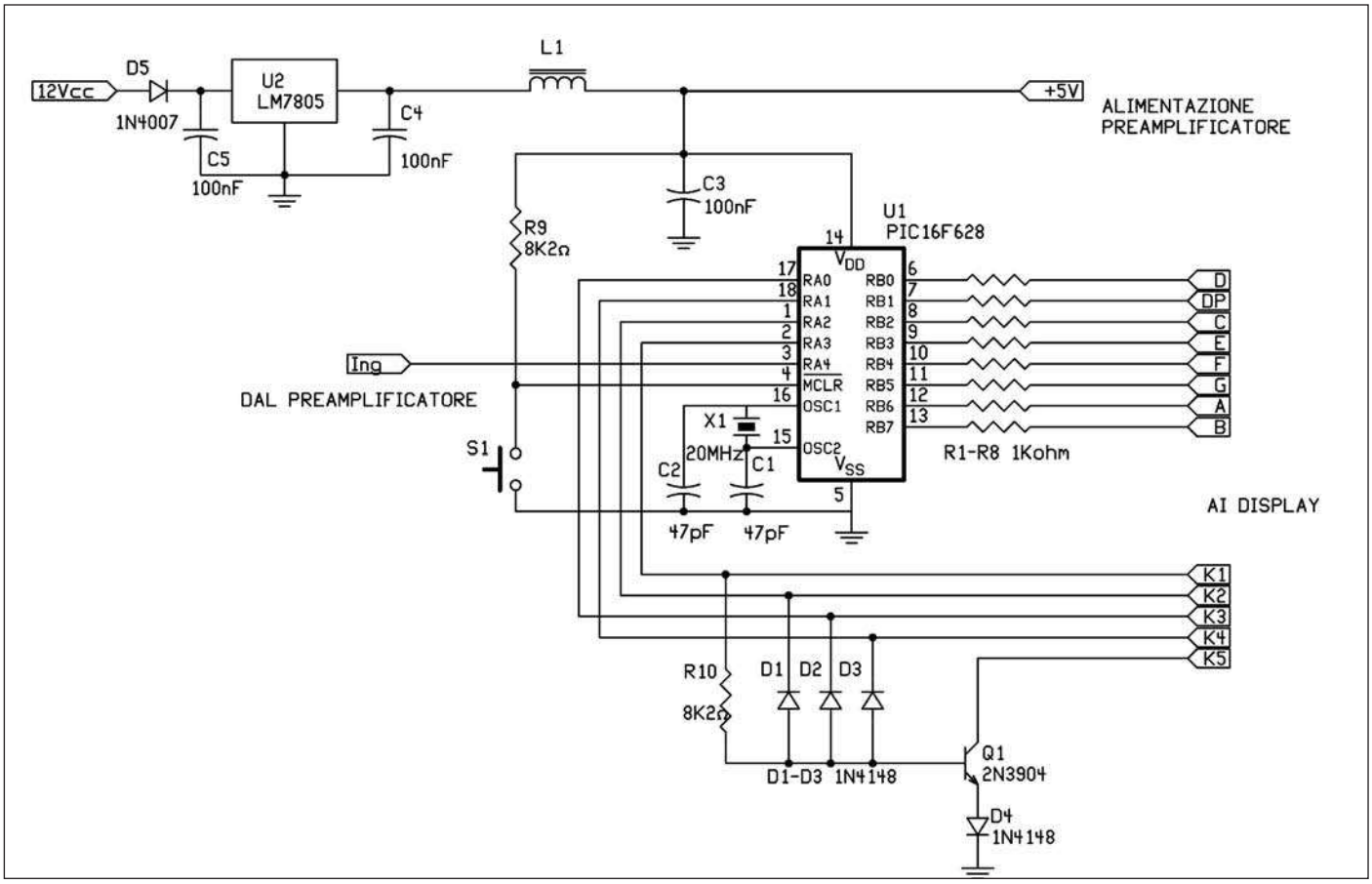


Figura 1: schema elettrico del frequenzimetro.

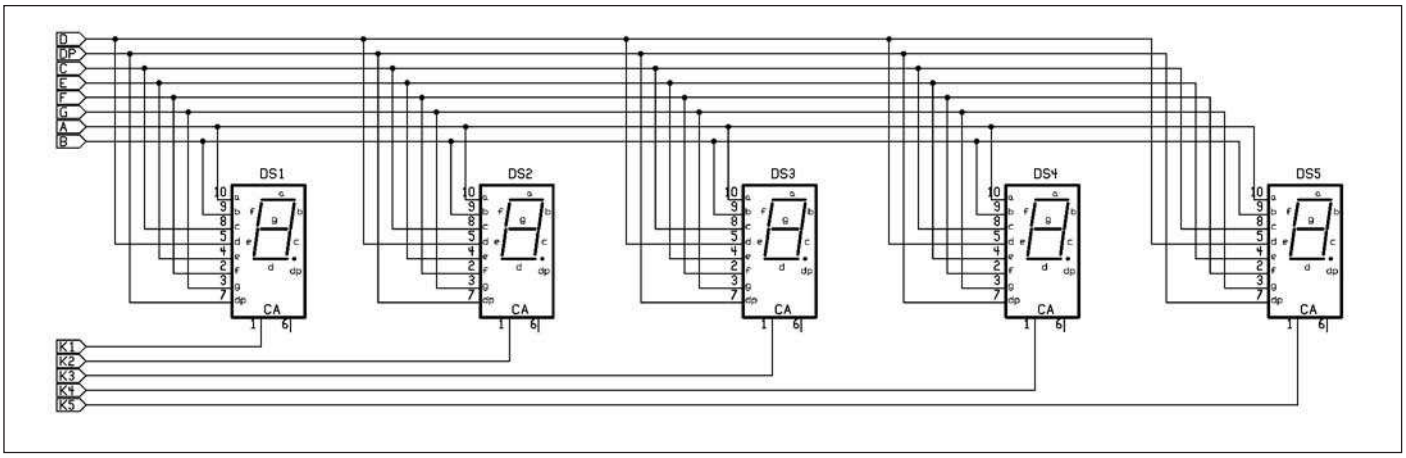


Figura 2: schema elettrico del display.

Durante le prove sul BiTx20, il ricetrasmettitore SSB in 20 metri, è emerso uno dei problemi maggiori di questo tipo di strumenti: generano moltissimo rumore. Nel caso specifico, la cosa è evidentemente aiutata dal fatto che il PIC funziona con clock a 4 MHz, esattamente la stessa frequenza del VFO della radio. Infatti per utilizzarlo sul citato ricetrasmettitore in 20 metri è stato successivamente realizzato questo modello derivato da un kit indiano, dal costo quasi nullo, 20 $ comprese le spese di spedizione, ed è basato su un progetto di DL4YHF, impiega un PIC 16F628, economico successore del noto 16F84, come il fratello minore è programmabile con un programmatore facile da realizzare anche per un uso saltuario. Anche il software di programmazione è reperibile in rete, icprog, ponyprog e altri.
Il kit è reperibile presso il gruppo indiano che propone anche il circuito stampato del ricetrasmettitore BiTx20 (http://www.cqbitx.blogspot.com). La scelta del progetto è stata decisa per la presenza dei display a sette segmenti a LED al posto del solito display LCD; non si tratta di risparmio sui componenti, ma della necessità di limitare lo spazio occupato dal display sul pannello del ricetrasmettitore, già piuttosto affollato.
La frequenza di funzionamento del microcontrollore, pari a 20 MHz, limita per quanto possibile i fischiettii e i rumoracci che di solito questi strumenti introducono nel ricevitore al quale sono collegati; esiste una versione del tutto analoga del firmware prevista per l'uso con un quarzo a 4 MHz. Per l'uso con il ricetrasmettitore è stato previsto un interruttore per spegnere il frequenzimetro in caso di necessità. Al frequenzimetro originale è stato aggiunto un preamplificatore a due stadi, il primo a FET a cui fa seguito un transistor NPN da commutazione. La frequenza di funzionamento dovrebbe avvicinarsi ai 60 MHz, in realtà per l'uso al quale è destinato non si è andati oltre gli 8 MHz, ma non vi è ragione per cui lo strumentino non debba rispettare i valori dichiarati. Il fatto che lo rende adatto all'impiego come sintonia elettronica è nella possibilità di effettuare la somma o la sottrazione del valore di media tra alcuni valori predefiniti e contenuti nella eeprom del PIC. Sono fornite le istruzioni per modificare i valori di media frequenza memorizzati nella eeprom del PIC, sono necessari una calcolatrice scientifica in grado di effettuare calcoli in esadecimale (quella di windows va bene) e un editor esadecimale (anche icprog o ponyprog); in seguito vedremo come procedere.
L'autore del firmware ha avuto una idea magnifica nell'inserire una routine che setta il valore di media a quello letto in ingresso. È dunque sufficiente inserire in ingresso al frequenzimetro l'uscita del BFO del ricevitore, oppure un generatore di segnali regolato sulla frequenza che ci interessa sommare (o sottrarre) al valore misurato, entrare in modalità setup per avere il nostro frequenzimetro programmato su valori anche strani e senza neppure aprirlo!
Fino ad oggi sono stati realizzati tre esemplari, due con sei display verdi da 7,6 mm (0.71"), di cui uno finto è utilizzato solo per completare frequenza di funzionamento della radio, l'altro con display analoghi a quelli impiegati molti anni fa sulle calcolatrici prodotte dalla Texas TI 5X; sono degli integrati in custodia DIL12 (6 + 6 pin) che contengono al loro interno tre display completi di punto decimale con i segmenti già collegati tra loro e i tre catodi disponibili all'esterno per la connessione al multiplex. Utilizzando dei display singoli è necessario realizzare tutti i collegamenti "a mano". In particolare, i segmenti andranno collegati tutti tra loro, ovvero tutti i segmenti "A" andranno collegati tra loro, così tutti i segmenti "B", fino a completare i sette segmenti e i punti (DP). Alla piastra del PIC andranno collegati i sette segmenti comuni a tutti i display e i catodi dei singoli display tenendo presente che "DS1" è il primo display a sinistra, mentre "DS5" è l'ultimo a destra. La sensibilità dello strumentino è più che buona, e comunque sufficiente per l'uso previsto; il VFO del ricetrasmettitore non viene caricato dall'inserzione dello strumentino grazie all'alta impedenza di ingresso fornita dal FET.

LO SCHEMA ELETTRICO

Il segnale di ingresso viene applicato al gate del FET attraverso un condensatore ceramico dal valore estremamente basso, questo per non caricare lo stadio a cui sarà collegato. Il FET impiegato è un canale N, è stato impiegato un vecchio 2N3819, ma anche un BF244 e dei vecchi FET della serie ECG. L'impiego di un modello più recente tipo il J310 è possibile e vantaggioso. La configurazione a drain comune fornisce una bassa impe-

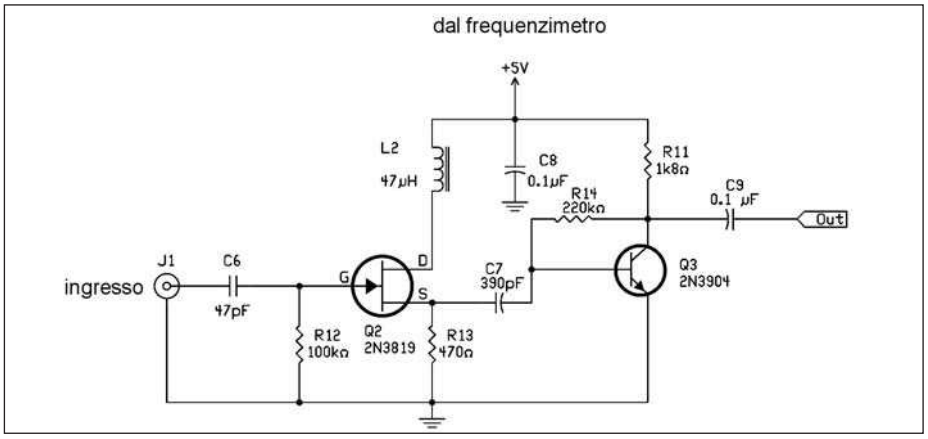


Figura 3: schema elettrico del preamplificatore.

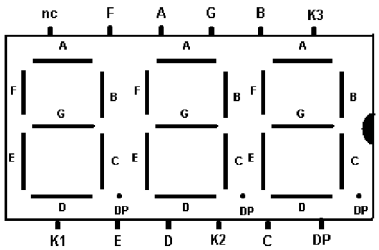


Figura 4: HP534S display 3 x 7 segmenti.

denza di uscita dello stadio che ben si adatta a quello successivo che è costituito da un transistor da commutazione (BC547 o 2N3904). Al fine di ottenere una buona sensibilità su tutta la gamma di interesse, è necessario curare il valore della resistenza di base del transistor, in quanto valori troppo bassi portano il transistor pericolosamente vicino alla zona di saturazione e impediscono che il transistor amplifichi come deve. Il risultato è una sensibilità estremamente ridotta. Se stiamo impiegando un transistor dalle buone caratteristiche in fatto di frequenza di taglio, quasi certamente avrà un guadagno in corrente più basso e la resistenza di base potrebbe scendere fino a 47 kohm, mentre se stiamo impiegando un transistor per commutazione come i due modelli citati prima, ma che hanno un guadagno estremamente alto (probabilmente vicino a 500 e oltre), è assolutamente necessario aumentare il valore della resistenza di base anche di molte volte, fino raggiungere i 220-470 kohm. In assenza di segnale, la tensione presente sulla base dovrebbe essere la metà della tensione di alimentazione, da misurarsi rigorosamente con un tester digitale, in quanto il vecchio Ice caricherebbe troppo lo stadio facendoci leggere un valore di tensione drammaticamente più basso del reale.

Questa versione utilizza display a catodo comune. È possibile impiegare display dotati di anodo comune capovolgendo i diodi D1, D2, D3 e D4; quest'ultimo andrà collegato a +5 V invece che a massa, sostituendo Q1 con un esemplare analogo a polarità invertita (un PNP, BC327 o analogo) e programmando il PIC con il file che DL4YHF ha chiamato "counter3.hex". L'ultimo display a destra (DS5) è stato aggiunto successivamente; dato che al PIC non avanzano più pin con cui comandare il quinto catodo, l'autore del firmware ha realizzato una porta OR a diodi permettendo di accendere l'ultimo display semplicemente quando tutti gli altri sono spenti. Il sistema funziona perfettamente, anche se all'accensione rimane per un attimo acceso il penultimo display, mentre l'ultimo è spento. Utilizzando comuni diodi al silicio (1N4148 o simili) è indispensabile la presenza di D4, altrimenti il transistor Q1 non commuta.

IL MONTAGGIO

Come sempre, i tre esemplari costruiti sono stati messi insieme su piastrina millefori; del resto le dimensioni e le esigenze dei contenitori utilizzati per i BiTx20 erano diverse, così i tre frequenzimetri. I componenti necessari sono veramente pochi e il montaggio su millefori non richiede più di una sera. Per i più pigri resta comunque la disponibilità del kit indiano; la velocità e la correttezza del gruppo citato è esemplare, in meno di tre settimane si ha tutto a casa con una spesa che è equivalente, se non qualche monetina in meno, a quel che costerebbero i soli componenti. Nella realizzazione pratica è bene rispettare alcuni consigli: l'ingresso del frequenzimetro andrà posto lontano dal quarzo e dall'alimentazione, che sarà ben filtrata facendo uso abbondante di impedenze e perline di ferrite per limitare i disturbi che il clock del PIC "porta" verso la radio. I collegamenti verso il display saranno realizzati con un connettori a stampato da 8+8 pin, il display sarà probabilmente fissato al pannello del ricetrasmettitore e la possibilità di separarlo dal resto è praticamente obbligatoria. Secondo l'avidità in corrente dei display impiegati, il regolatore di tensione potrebbe avere bisogno di un piccolo radiatore, quindi sulla piastrina millefori ne è stato previsto lo spazio, anche se in realtà non si è dimostrato necessario.

Il settaggio dello strumento avviene con un pulsante. È bene collocarlo a stampato, non è necessario averlo sul pannello anteriore, a meno che non lo si utilizzi come strumentino a sé stante. Una posizione sul lato della piastra, possibilmente verso un lato del contenitore, oppure sul lato inferiore, permette di accedervi con un piccolo cacciavite semplicemente praticando un foro al punto giusto nel contenitore che non dovrà neppure essere aperto.

Il collegamento verso l'ingresso sarà realizzato con un cavetto schermato adatto, il solito RG174 è perfetto. Si evita l'impiego di cavetti schermati per bassa frequenza, in quanto il frequenzimetro genera già abbastanza rumore da solo.

Le due impedenze L1 e L2 sono un ulteriore blocco alla radiofrequenza; è possibile impiegare due piccoli nuclei toroidali su cui avvolgeremo alcune spire di filo da 0,7-0,8 mm. Il valore di induttanza non è assolutamente critico, lo strumento funziona bene anche senza, ma la loro presenza riduce ulteriormente il rumore generato, o almeno gli impedisce di "viaggiare" sfruttando i collegamenti delle alimentazioni. In alternativa delle piccole impedenze su ferrite anche di recupero andranno ugualmente bene.

Probabilmente il quarzo a 20 MHz che fornisce il clock allo strumentino avrà un robusto scarto dalla frequenza nominale. La taratura del frequenzimetro av-

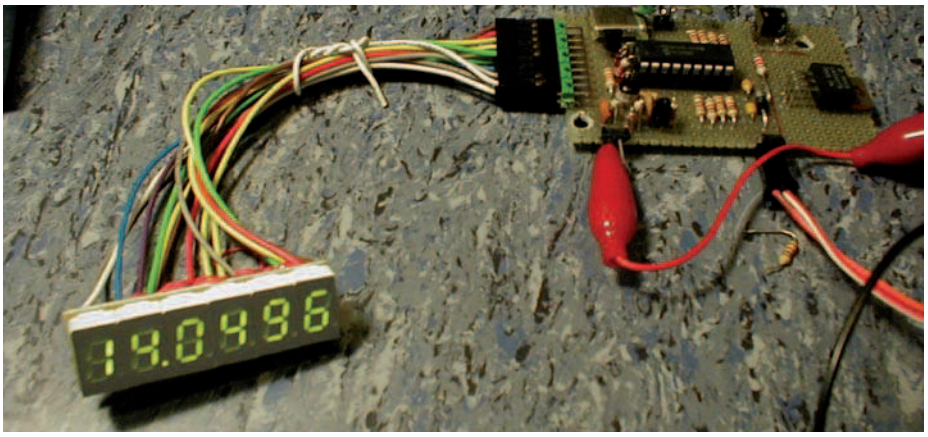


Figura 5: il frequenzimetro montato.

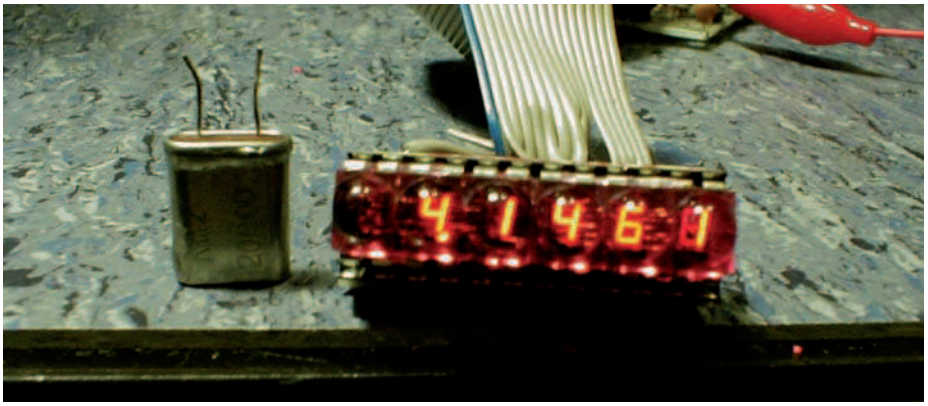


Figura 6: il display piccolo.

viene variando il valore delle due capacità collegate tra il quarzo e massa, C1 e C2, il cui valore andrà aumentato o diminuito secondo necessità. I due condensatori dovrebbero essere uguali, ma se in fase di taratura vediamo che la variazione del valore di un solo condensatore ci ha portato ad avere la lettura precisa, possiamo tranquillamente lasciare le cose come sono. Per tarare il frequenzimetro è stato utilizzato un altro esemplare "vecchio stile"; l'uso come indicatore di sintonia ha portato a ritenere valida una misura con uno scarto di meno di 100 Hz rispetto alla lettura del frequenzimetro "campione". In alternativa è possibile, se il tutto è utilizzato su un ricetrasmettitore, effettuare delle prove comparative tra un segnale ricevuto con un rtx commerciale e il nostro piccolino.

CAMBIARE IL VALORE DELLA MEDIA

Il file originale, con cui dovremmo programmare il PIC (ricordo per display a catodo comune) è "counter2.hex" diffuso da DL4YHF e reperibile in rete e sul sito della rivista. Nel caso in cui avessimo necessità di modificare i valori preimpostati nella eeprom del PIC è necessario munirci di calcolatrice scientifica e di editor esadecimale (i citati icprog o ponyprog vanno bene). Dobbiamo leggere la frequenza del BiTx americano il cui VFO lavora tra 3.100 e 3.280 circa. Nella versione indiana il VFO è più alto di 1 MHz ed essendo la media a 10 MHz basta leggere la frequenza del VFO, aggiungere mentalmente 10 MHz e ottenere la frequenza di funzionamento del ricetrasmettitore. Oppure aggiungere un "1" fisso a sinistra del display per ottenere la lettura falsa, ma apparentemente completa.
Purtroppo sulla versione americana il valore della media a 11 MHz ci impedisce di sfruttare lo strumento per la lettura fino ai 100 Hz; se sommiamo correttamente gli 11 MHz, otteniamo la lettura da 14.100 a 14.280, impiegando le 5 cifre utili. Se al contrario sommiamo solamente 1 MHz, otteniamo la lettura da 4.1000 a 4.2800, includendo quindi la lettura fino alle centinaia di Hz, ritorniamo dunque alla situazione appena vista e possiamo aggiungere a sinistra del display il solito "1" finto a esclusivo beneficio estetico.
Torniamo a noi, con la calcolatrice scientifica convertiamo in esadecimale il valore, espresso in Hz, da sommare alla lettura dello strumento: 1 MHz, ovvero 1.000.000 Hz, in esadecimale diventa F4240, ed è il valore che andremo a porre alla locazione di memoria 000800, la cui riga diventerà 00 00 00 00 0F 42 40.
Se avessimo voluto sottrarre lo stesso valore avremmo dovuto calcolarne, sempre in esadecimale, la differenza con FFFFFFFF, ovvero:

FFFFFFFF – F4240 = FFF0BDBF

LISTA COMPONENTI

R1	1 kohm
R2	1 kohm
R3	1 kohm
R4	1 kohm
R5	1 kohm
R6	1 kohm
R7	1 kohm
R8	1 kohm
R9	8.2 kohm
R10	8.2 kohm
R11	1.8 kohm
R12	100 kohm
R13	470 ohm
R14	220 kohm vedi testo
C1	47 pF
C2	47 pF
C3	100 nF
C4	100 nF
C5	100 nF
C6	47 pF
C7	390 pF
C8	100 nF
C9	100 nF
D1	1N4148
D2	1N4148
D3	1N4148
D4	1N4148
D5	1N4148
X1	Quarzo 20 MHz
L1	RFC 47 – 100 microH
L2	RFC 47 – 100 microH
U1	PIC16F628 a 20 MHz
U2	LM7805
Q1	2N3819, BF244, J310
Q2	2N2904, BC547c
Q3	2N2904, BC547c
S1	Pulsante da CS NA
DS1-DS5	5 display a LED a catodo comune, oppure due chip HP534S, che contengono tre display l'uno
	Zoccolo a 18 pin DIL per il PIC
	Connettori per il display

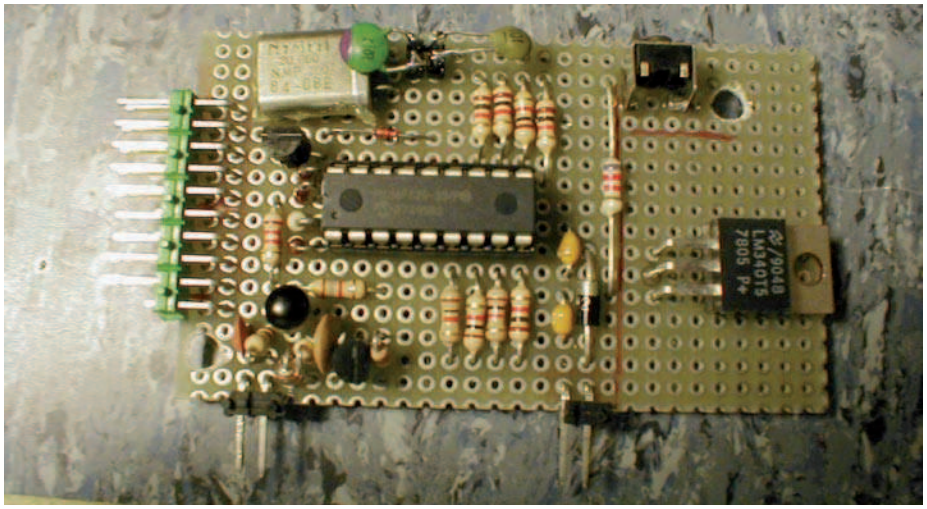


Figura 7: il frequenzimetro montato su millefori.

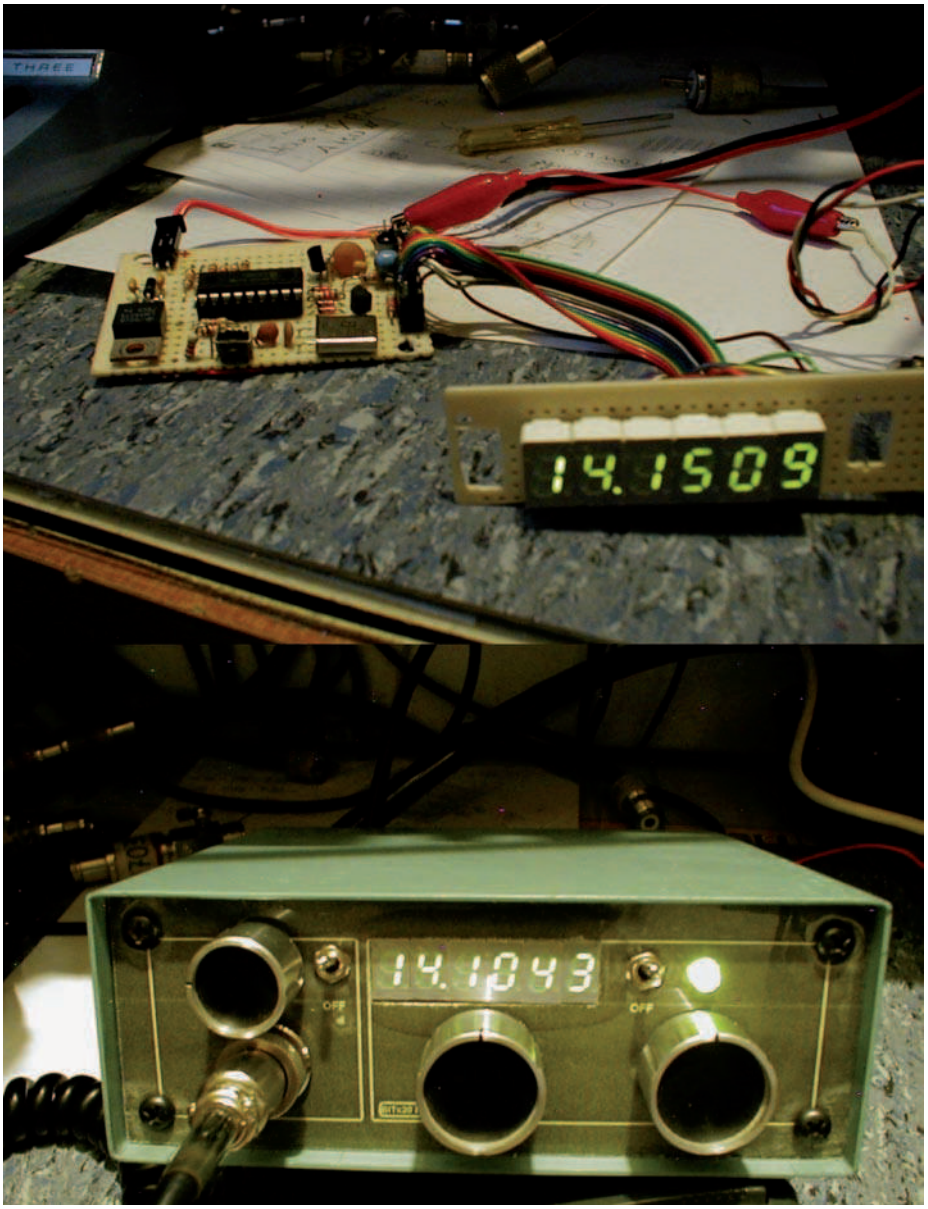


Figura 8: un altro esemplare montato e il BiTx20 che lo ospita.

La sequenza da inserire nella eeprom diventa quindi 00 00 00 FF F0 BD BF. Se intendiamo aggiungere al frequenzimetro il sesto display "finto", che andrà posizionato a sinistra del primo display (DS1) dello strumento, è necessario accendere i soli segmenti "b" e "c", in modo da formare un "1" fisso che completa la frequenza di funzionamento dell'RTX e trasforma l'eventuale lettura da 4.0496 ai più gradevoli 14.0496. In serie ai due segmenti andranno poste due resistenze il cui valore dovrà essere poco superiore a quello di R1 - R8 (quindi 1,2 – 1,5 kohm) al fine di ottenere un display con una luminosità analoga ai sui fratelli. Nella EEPROM sono inseriti di default alcuni valori di media frequenza, disponibili per essere sommati, o sottratti, dalla frequenza presente all'ingresso. Sono selezionabili dal menu in modalità SET.

MENU DI SETTAGGIO

Il tasto SET ha due funzioni: tenendolo premuto a lungo si entra nel menu visualizzato, mentre una pressione breve del tasto permette di scorrere le scelte disponibili che vanno confermate con una pressione lunga dello stesso tasto. Nel complesso, pur essendo la rappresentazione sul display a 7 segmenti forzatamente limitata, l'uso del menu è intuitivo e veloce. Ecco le principali voci del menu:

- *Quit:* esce dal menu di programmazione senza modificare nulla;
- *Add:* somma la frequenza misurata al valore stabilito della media frequenza;
- Sub: sottrae dalla frequenza misurata il valore stabilito della media frequenza;
- *Zero:* setta la lettura senza alcun offeset, viene visualizzata la frequenza realmente misurata senza intervenire sul suo valore;
- *Table:* permette di scegliere tra i valori di media predefiniti, ovvero 455 kHz, 4.1943 MHz, 4.4336 MHz, 10.700 MHz, o l'eventuale valore che abbiamo aggiunto con la procedura appena descritta;
- *PSave-NoPSV:* permettere di spegnere il display dopo 15 secondi in cui lo strumento non ha rilevato una variazione della frequenza di ingresso; l'impiego di questa possibilità è utile in unione a RTX a bassa potenza alimentati a batterie in cui il consumo del solo display è probabilmente paragonabile a quello del ricevitore.

Per una lettura gradevole e "sensata" dello strumento è necessario non abusare della possibilità di sottrarre il valore di media. È bene che il risultato prevedibile della nostra sottrazione non sia troppo vicino allo zero. Se la frequenza misurata "vera" è vicina al valore della media selezionato, rischiamo di far entrare in funziona l'*autoranging* dello strumento che seleziona portate più basse. Se sul BiTx statunitense avessimo utilizzato la funzione di sottrazione del VFO sottraendo 3.000 MHz al fine di ottenere una maggior precisione di lettura, avremmo potuto trovarci in questa situazione: il VFO funziona da 3.100 a 3.280, se sottraiamo 3 MHz la lettura andrà da 100 a 280 kHz, sul display si leggono i valori compresi tra 100.00 e 280.00, ovvero la lettura si estende fino alle decine di Hertz. Perdiamo l'estetica di avere la frequenza completa a favore di una maggior precisione. I problemi giungono appena la frequenza del VFO scende ancora un poco sotto i 3.100 MHz, a 3.090 il frequenzimetro legge 90 kHz, che visualizza come 90.000; la lettura degli Hertz sarà certamente molto instabile e poco attendibile e il risultato estetico è sgradevole, ancor più se allo strumento sono state aggiunte cifre "finte" al fine di completare la lettura della frequenza. A questo proposito DL4YHF consiglia di mantenere il risultato della sottrazione ad almeno 120 kHz, per avere il piccolo margine di sicurezza prima dell'intervento dell'*autoranging*.
Lo strumento funziona su cinque possibili scale, ovvero:

- 1-9.999 kHz con un periodo di lettura pari a 1 secondo;
- 10-99.990 kHz con un periodo di lettura pari a 0,5 secondi;
- 100-999,90 kHz con un periodo di lettura pari a 0,25 secondi.

In queste tre prime portate il punto decimale lampeggia a significare che sono visualizzati dei kHz:

- 1-9.9990 MHz con un periodo di lettura pari a 0,25 secondi;
- 10-60.000 MHz con un periodo di lettura pari a 0,25 secondi.

In queste ultime due portate il punto è fisso e la visualizzazione indica dei MHz. In tutte le portate, salvo che nella prima, la visualizzazione avviene su cinque cifre aumentando di volta in volta la portata del display meno significativo. Da questa tabella appare evidente che la misura del VFO di un ricetrasmettitore a cui è sottratto il valore della media frequenza non deve assolutamente cadere nei pressi di un cambio di scala, pena il cambio di precisione e il conseguente spostamento delle cifre verso sinistra, che noi potremmo interpretare come 900 kHz al posto di 90 kHz. ❑

CODICE MIP 2757047

di GIUSEPPE MODUGNO

Lettore di SMARTCARD

(parte seconda)

In questa seconda e ultima parte riprendiamo il progetto del lettore stand-alone di smartcard dotato di display e tastierino per l'utilizzo in applicazioni di fidelizzazione della clientela. Dopo aver realizzato il prototipo completo, andremo qui a discuterne il funzionamento e il firmware che lo anima

Partiamo dalla descrizione del funzionamento del lettore di smartcard, senza entrare nei dettagli tecnici. Più avanti, nel prossimo paragrafo, verrà fatta una descrizione dettagliata del firmware del microcontrollore.

SCHERMATA DI BENVENUTO

```
   Benvenuto!
Iniz. in corso..
```

È semplicemente una schermata che viene visualizzata, per qualche secondo, subito dopo l'accensione.

PRIMA ACCENSIONE

Alla prima accensione del lettore, dopo la schermata di benvenuto, viene visualizzata quella che permette di impostare il PSC da utilizzare per la scrittura delle smartcard.

```
Old PSC: FFFFFF
New PSC:
```

Il PSC di default (prima riga) è FFFFFF. Il cursore lampeggia in corrispondenza della prima cifra del nuovo PSC (seconda riga). Premere i tasti corrispondenti alle sei cifre del PSC desiderato. A ogni digitazione il cursore si sposta alla cifra successiva. Dopo aver digitato sei cifre, il cursore lampeggia ancora sull'ultimo carattere del display attendendo la conferma (tasto F) oppure l'annullamento dell'operazione (tasto E). Annullando l'operazione, il lettore continuerà a utilizzare il PSC di default. Questa schermata viene visualizzata solo alla prima accensione. Per modificare successivamente il PSC è necessario andare nella schermata setup.

SCHERMATA SETUP

```
1. Imposta PSC
2. Contatori
```

Questa schermata viene visualizzata quando si tiene premuto il tasto F durante l'accensione del lettore. È una schermata di servizio che non dovrebbe essere conosciuta dal normale utilizzatore dell'apparecchio, poiché consente di modificare alcune impostazioni fondamentali, tra cui lo stesso PSC. Premere il tasto 1 per passare alla schermata di impostazione del PSC (già vista precedentemente), il tasto 2 per passare alla schermata relativa ai contatori, il tasto E per tornare alla schermata principale.

SCHERMATA CONTATORI

Il lettore memorizza il numero delle operazioni relative agli accessi alle smartcard inserite. Ci sono quattro contatori per altrettante tipologie di operazioni: letture corrette (aaaa), letture errate (bbbb), scritture corrette (cccc), scritture errate (dddd). Ogni contatore può arrivare sino a un massimo di 9999, dopodiché non si incrementa più. In questa stessa schermata è possibile azzerare tutti i contatori contemporaneamente, premendo il tasto 0. Premere il tasto E per tornare alla schermata di setup.

```
Lt OKaaaa Erbbbb
Sc OKcccc Erdddd
```

SCHERMATA PRINCIPALE

La schermata principale visualizza semplicemente un messaggio che chiede di inserire una carta. È la schermata che viene normalmente visualizzata dopo la schermata di benvenuto (eccetto alla prima accensione).

```
Inserire
card
```

SCHERMATA DATI

```
Giuseppe Modugno
Credito: 100,00E
```

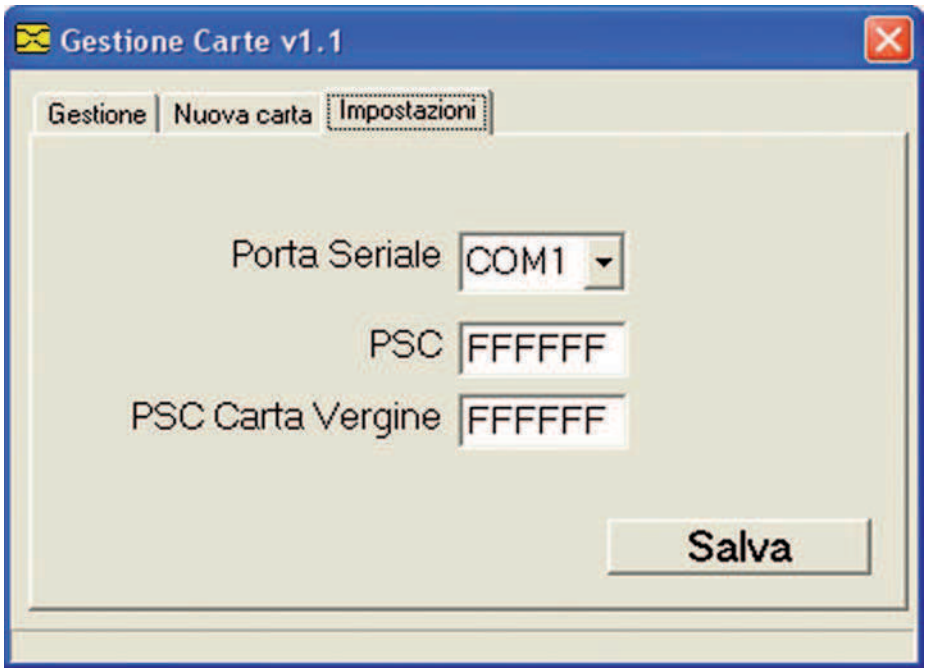


Figura 1: finestra relativa alle impostazioni del programma "Gestione Carte".

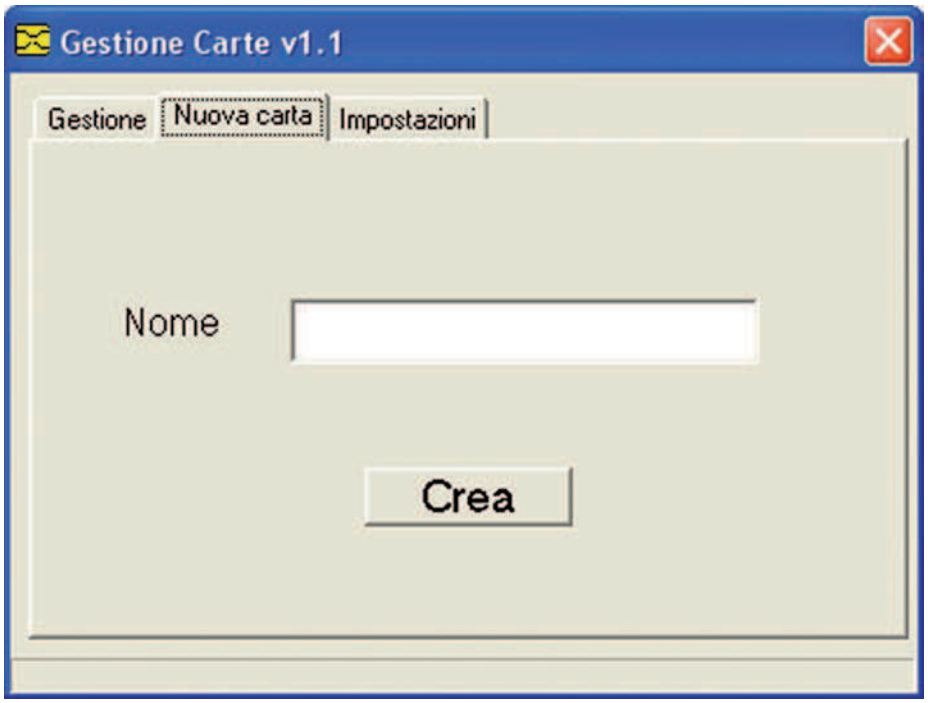


Figura 2: finestra relativa alla creazione di una nuova carta.

Non appena viene inserita una smartcard valida all'interno del lettore, si passa automaticamente alla schermata dati. Se viene inserita una smartcard non valida, viene visualizzato un messaggio d'errore fino a quando la carta non viene estratta. Sulla prima riga viene visualizzato il nome del proprietario della carta, sulla seconda il credito associato a quella carta (il carattere E finale sta a indicare "euro", supponendo che il credito sia un equivalente in moneta, ma potrebbe indicare qualsiasi altra cosa: punti, gettoni, ecc). Premere il tasto F per modificare il credito. Un cursore lampeggiante compare sulla prima cifra del credito a indicare che è possibile premere i tasti numerici per modificarne il valore. Durante la modifica del credito è possibile premere, in qualsiasi momento, il tasto E per annullare l'operazione. Dopo aver premuto cinque cifre, il cursore lampeggia in corrispondenza della lettera E, in attesa di una conferma (tasto F) o di un annullamento (tasto E) da parte dell'utente. Confermando la modifica con il tasto F, il lettore proverà a memorizzare il nuovo valore nella smartcard. Se l'operazione ha successo, il LED verde si accende. Se l'operazione non ha successo, si accende il LED rosso e viene visualizzato un messaggio d'errore per qualche secondo. A seconda del risultato delle operazioni di lettura e scrittura, i contatori vengono di conseguenza aggiornati. Per ritornare alla schermata principale, è sufficiente estrarre la smartcard.

INIZIALIZZAZIONE DI UNA SMARTCARD

Il nostro lettore è, come già detto, compatibile solo con alcune tipologie di smartcard, le SLE4442. Prima di utilizzare una smartcard vergine con il lettore, è necessario inizializzarla in modo da renderla compatibile con il sistema. A tale scopo, è necessario utilizzare un altro lettore di smartcard collegato a un computer e un software apposito. Il lettore da collegare al computer tramite connessione RS232 è denominato UniReader ed è stato progettato in un articolo pubblicato su Fare Elettronica (in particolare sul numero 231 di settembre 2004). Il software da utilizzare sul PC per comunicare con l'UniReader e inizializzare la smartcard vergine è "Gestione Carte", sviluppato e descritto sempre su un precedente articolo di Fare Elettronica (numero 232 di ottobre 2004). In realtà, il software è stato leggermente modificato rispetto all'edizione originale per adattarlo al nuovo sistema. È possibile scaricare sia il sorgente (in Microsoft Visual Basic 6.0) che il file di installazione dal sito di Fare Elettronica. All'avvio del programma "Gestione Carte" sul computer, cliccate sulla linguetta "Impostazioni" (**figura 1**). Qui è possibile impostare la porta seriale utilizzata per il collegamento con l'UniRea-

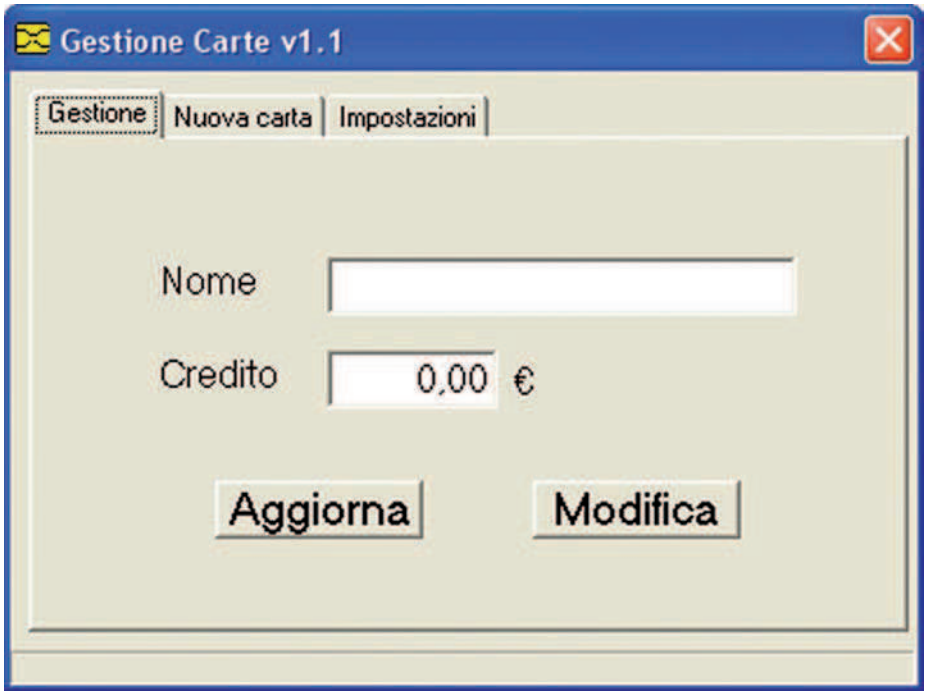


Figura 3: finestra relativa alla gestione di una carta già inizializzata.

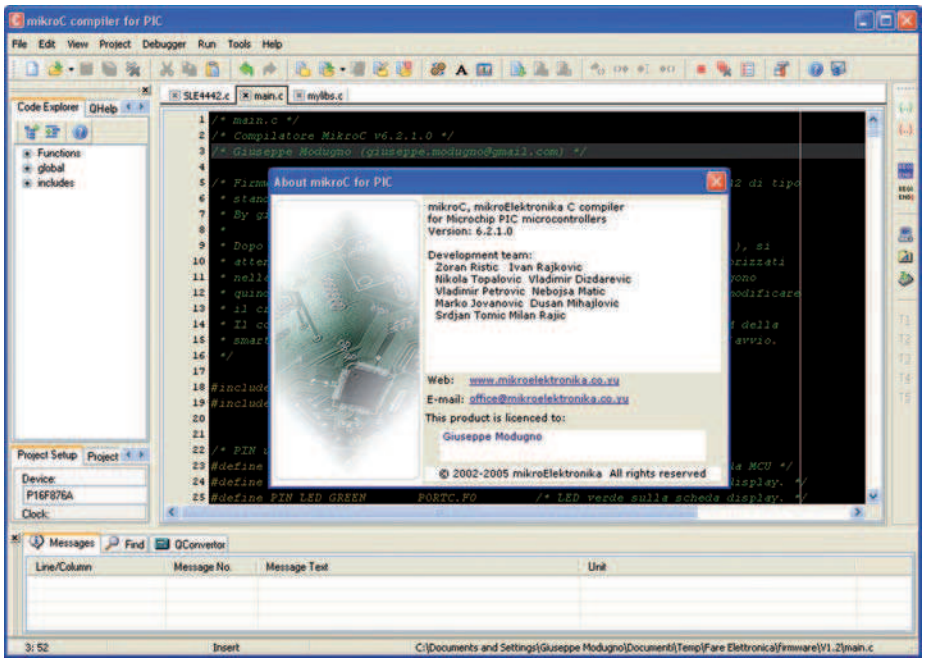


Figura 4: screenshot dell'ambiente di sviluppo mikroC utilizzato per questo progetto.

der, il PSC della smartcard vergine e il nuovo PSC da impostare e che sarà utilizzato durante l'applicazione. Dopo aver scelto questi parametri (controllate bene il valore dei due PSC), cliccare sul tasto "Salva" per memorizzare permanentemente le modifiche in modo che vengano caricate automaticamente alla prossima esecuzione del software. È possibile passare ora alla finestra "Nuova Carta" (**figura 2**), dove inseriremo il nome del titolare della smartcard (massimo 16 caratteri). Cliccando sul tasto "Crea" verrà inizializzata la smartcard inserita nell'UniReader. Se l'operazione non va a buon fine, è necessario controllare meglio il collegamento con l'UniReader e il valore del PSC relativo alla smartcard vergine impostato nella finestra "Impostazioni". È importante verificare il buon funzionamento dell'UniReader (si faccia riferimento all'articolo di Fare Elettronica già citato).
Dopo aver inizializzato la smartcard, è possibile utilizzarla con il nostro lettore standalone per applicazioni di fidelizzazione. Volendo, è possibile leggere il contenuto della smartcard ed eventualmente modificarlo, utilizzando la schermata "Gestione" del software (**figura 3**). Cliccare sul tasto "Aggiorna" per leggere il contenuto della smartcard inserita nell'UniReader e visualizzarlo nella finestra. Cliccare su "Modifica" per scrivere i dati mostrati a video nella smartcard inserita nell'UniReader.

FIRMWARE

È ovvio che in circuiti come questi, oltre all'hardware, anche il software ha un ruolo importante e crediamo che sia molto utile spiegarne, nel modo più dettagliato possibile, il funzionamento con codice sorgente alla mano. Nei prossimi paragrafi cercheremo di commentare ampie porzioni del codice sorgente, senza per questo scorrere in modo arido le linee di codice dall'inizio alla fine. Di volta in volta prenderemo in considerazione un aspetto e commenteremo le zone del codice (non necessariamente consecutive nel file sorgente) che rendono possibile tale funzionalità. Per una maggiore leggibilità dei listati relativi al codice sorgente, alcuni commenti sono stati eliminati o ridotti.

LINGUAGGIO E AMBIENTE DI SVILUPPO

Per applicazioni molto semplici, il linguaggio di riferimento è sicuramente l'Assembler. Per la nostra applicazione, che deve gestire contemporaneamente un display, un tastierino numerico e una smartcard, la scelta obbligata è il linguaggio ad alto livello C. Tra i diversi compilatori C e ambienti di sviluppo per i microcontrollori Microchip, abbiamo scelto l'ottimo mikroC della mikroElektronika (www.mikroe.com). Noi siamo in possesso e abbiamo usato la versione 6.2.1.0 (**figura 4**).
La procedura di installazione e l'utilizzo sono veramente semplici e non è necessario entrare nel merito in questo articolo. Per chi non voglia cimentarsi con la compilazione del codice sorgente e non abbia interesse a modificarne il funzionamento, è sufficiente prelevare il file firmware.hex

Sistemi di registrazione... discreti

Orologio con telecamera e registratore A/V integrati

Dispone di microtelecamera CMOS nascosta dentro la cifra 2 e consente di effettuare fino a 150 minuti di registrazioni audio/video con risoluzione di 352x288 pixel (CIF). Integra una memoria flash da 2GB sulla quale vengono salvati i filmati, scaricabili su PC mediante collegamento USB.

Penna con telecamera e registratore A/V integrati

È in grado di registrare filmati audio-video in alta risoluzione (15÷30fps), tramite la microtelecamera nascosta, solamente premendo un pulsante. Le registrazioni in formato AVI vengono salvate direttamente sulla flash memory (da 2GB) di cui dispone e possono essere trasferite ad un PC tramite collegamento USB. Batteria ricaricabile integrata.

... e per applicazioni speciali.

Mini DVR con telecamera a colori e monitor TFT

Compatto ed affidabile, integra un monitor LCD TFT da 2" e dispone di una telecamera pin-hole a colori, completa di microfono. Le immagini riprese vengono salvate direttamente sulla mini SD Card (max. 4GB) inserita nello slot e possono essere trasferite ad un PC con un semplice "Copia e Incolla". Formato di registrazione AVI; rsoluzione 640x480; ingresso e uscita A/V; batteria ricaricabile ai polimeri di litio integrata; autonomia in registrazione ~2h; dimensioni monitor 66 x 55 x 17mm, telecamera 34 x 22 x 17mm. Mini SD Card da 2GB inclusa.

CP383

€ 208.00

Camera CAR a colori

€ 99.00

Progettata per essere montata sul cruscotto di un auto, questa camera car a colori, permette di catturare sequenze video.
Utile ai fini assicurativi o per puro divertimento!
Dispone di un puntatore laser che permette di allineare l'obiettivo della telecamera alla strada da riprendere. Utilizzando un cavo AV è possibile collegare il dispositivo ad un monitor con ingresso videocomposito per visualizzare il video in real-time.
Sensore d'immagine CMOS a colori; risoluzione video 320 x 240 (QVGA); angolo di ripresa 60°; formato di registrazione AVI; registrazione su SD/MMC Card 2GB max. (non inclusa); dimensione file video 4~8MB/minuto; alimentazione a batterie (3 x AAA) o 12Vdc tramite adattatore per presa accendisigari (incluso); consumo max. 220mA; dimensioni 104 x 48 x 20mm.

CAMCOLVC3

- **Guarda il video su Youtube:**
http://www.youtube.com/watch?v=zcisHI1zvhU

CODICE MIP 2753569

FUTURA ELETTRONICA idee in elettronica
Via Adige, 11 • 21013 Gallarate (VA)
• Tel. 0331/799775 • Fax. 0331/792287

Maggiori informazioni su questi prodotti e su tutte le altre apparecchiature sono disponibili sul sito *www.futurashop.it* tramite il quale è anche possibile effettuare acquisti on-line.

Tutti i prezzi sono da intendersi IVA inclusa.

LISTATO 1
Estratto di codice per la gestione del display

```
...
const char str_welcome1[]
= "   Benvenuto!   ";
const char str_welcome2[]
= "Iniz. in corso..";

void
main( void )
{
...
     /* Inizializzazione
LCD. */
      Lcd_Custom_Config(
&PORTC, 7, 6, 5, 4,

&PORTC, 2, 2, 3 );

Lcd_Custom_Cmd(LCD_CUR-
SOR_OFF);

     /* Schermata intro-
duttiva. */
     myLcd_Custom_Out( 1,
1, str_welcome1 );
     myLcd_Custom_Out( 2,
1, str_welcome2 );
...
}

void
myLcd_Custom_Out(
    unsigned short row,
    unsigned short col,
    const char *txt )
{
    strcp_c(strbuf, txt);
     Lcd_Custom_Out( row,
col, strbuf );
}
```

dall'archivio firmware.zip che troverete sul sito di Fare Elettronica alla pagina relativa a questo articolo. Utilizzate tale file per la programmazione del microcontrollore (si veda l'articolo precedente).
Per i più esperti, invece, si consiglia di scompattare tutti i file dell'archivio in una cartella vuota e aprire, con mikroC, il file firmware.ppc. Come potrete vedere, il firmware è formato da 3 moduli C:

- mylibs.c;
- SLE4442.c;
- main.c;

e 2 file include .h:

- mylibs.h;
- SLE4442.h.

mylibs.c

È sempre buona norma cercare di suddividere un programma complesso in tanti piccoli sotto-processi, mediante un'operazione ricorsiva, fino ad arrivare a processi minimali facilmente codificabili attraverso una funzione (nel caso di linguaggi procedurali, come il linguaggio C). Nel modulo mylibs.c sono presenti alcune funzioni di base utilizzate un po' dovunque in altri punti del sorgente. Per esempio, la funzione *strcp_c()* copia una stringa costante dalla Flash nella memoria RAM. Questa funzione è necessaria soprattutto per la scrittura di stringhe sul display, come si vedrà in seguito. La funzione *hex()* che utilizza, a sua volta, la *ushort2hex()*, converte un array di byte in una stringa che ne rappresenta il contenuto in esadecimale. Questa funzione viene utilizzata principalmente per la visualizzazione del PSC della smartcard sul display.
La funzione checksum() calcola un valore di controllo utilizzato per verificare la coerenza dei dati memorizzati nella smartcard inserita nel lettore. Questo valore viene calcolato semplicemente come somma binaria (operazione XOR) di tutti i byte su cui si vuole calcolare la checksum. Se uno dei byte dovesse essere modificato (volontariamente da un pirata o involontariamente per un malfunzionamento), senza modificare anche la checksum, il lettore può accorgersi facilmente che i dati memorizzati sono corrotti e può rifiutare la smartcard inserita.
Infine le *myEEPROM_Read()* e *myEEPROM_Write()* leggono e scrivono nella EEPROM interna del microcontrollore un array di byte di lunghezza variabile, richiamando le funzioni di libreria *EEPROM_Read()* e *EEPROM_Write()*, messe a disposizione dal compilatore mikroC, che si limitano a leggere/scrivere un singolo byte. Tutte queste funzioni sono molto semplici e non pensiamo sia il caso di appensantire la trattazione commentandone il codice relativo.

SLE4442.c

Il modulo SLE4442, così come fa intendere il nome, contiene tutte le funzioni che riguardano la comunicazione con la smartcard. Sono tutte funzioni il cui nome inizia per *mc* che sta per memory card. Non entriamo nei dettagli di questo modulo, poiché il codice contenuto all'interno è esattamente lo stesso di quello proposto nel n. 232 dell'ottobre 2004 di Fare Elettronica, nell'articolo intitolato "Smartcard (quinta parte): applicazioni con le smartcard a memoria".

main.c

Il modulo main.c è ovviamente il più importante, poiché rappresenta il programma principale che gestisce tutte le nostre periferiche: smartcard, tasti, LED, display, ecc.

Display

La gestione del display è affidata quasi completamente alle librerie già pronte presenti nel compilatore mikroC. In particolare, ci stiamo riferendo alla "LCD Custom Library" di cui potete leggere nell'Help on-line. Con la funzione *Lcd_Custom_Config()* indichiamo alla libreria quali sono i pin del microcontrollore utilizzati per la comunicazione con il display. Anche se questa funzione vuole conoscere il pin relativo al segnale R/W, in modalità 4 bit questo segnale non è usato e quindi non ha importanza.
La funzione *Lcd_Custom_Cmd()* permette di inviare al display un comando per il controller HD44780 (per esempio, cancella il contenuto del display, sposta il cursore sulla prima riga, abilita la visualizzazione del cursore lampeggiante, ecc).
La funzione *Lcd_Custom_Out()* è forse la più importante, poiché permette di scrivere una stringa sul display alla posizione voluta. Però bisogna stare attenti a utilizzare questa funzione, poiché essa accetta, come stringa, un puntatore a una locazione in RAM. Se avessimo dovuto memorizzare tutte le stringhe di questa applicazione in RAM, avremmo esaurito molto presto l'esigua capacità di questa memoria, che è di soli 368 byte. È per questo che abbiamo creato una nuova funzione, la my*Lcd_Custom_Out()*, che accetta come stringa un puntatore nella memoria Flash (*const char **), dove sono conservate tutte le stringhe necessarie. Prima di richiamare la funzione di libreria *Lcd_Custom_Out()*, la my*Lcd_Custom_Out()* copia la stringa dalla Flash in RAM, utilizzando sempre lo stesso spazio in RAM (l'array *strbuf[]*), quindi senza spreco. Naturalmente per copiare

LISTATO 2
Estratto di codice per la gestione del tastierino

```
const unsigned short keytable[4][4] = {
  /* RB4, RB5, RB6, RB7 */
    {0xC, 0xB, 0x0, 0xA},       /* RB0 */
    {0xD, 0x3, 0x2, 0x1},       /* RB1 */
    {0xE, 0x6, 0x5, 0x4},       /* RB2 */
    {0xF, 0x9, 0x8, 0x7},       /* RB3 */
};
...
/* Tasto premuto (0xFF x nessun tasto premuto). */
unsigned short key = 0xFF;
...
void
interrupt( void ) {
    static unsigned short rowmask=0x01;
    static unsigned short row = 0;
    static unsigned short tmr_isr = 0;
    unsigned short port;

    port = ~((PORTB>>4)|0xF0);

    if( !tmr_isr ) {
        tmr_isr = 20;
        if( port&0x01 )
            key = keytable[row][0];
        else if( port&0x02 )
            key = keytable[row][1];
        else if( port&0x04 )
            key = keytable[row][2];
        else if( port&0x08 )
            key = keytable[row][3];
        else
            tmr_isr = 0;
        ...
        if( ++row==4 ) {
            row=0;
            rowmask=0x01;
        } else
            rowmask<<=1;
        PORTB = ~rowmask;
    } else
        tmr_isr–;

    PIR1.TMR1IF = 0;
}
...
void
main( void )
{
...
    T1CON = 0b00000001;
    PIE1.TMR1IE = 1;
    INTCON.PEIE = 1;
    INTCON.GIE = 1;
...
}
```

LISTATO 3
Codice relativo alla gestione della EEPROM

```
struct {
  char sign[5];
  unsigned short psc[3];
  unsigned int cnt_readok;
  unsigned int cnt_readbad;
        unsigned    int
cnt_writeok;
  unsigned int cnt_writebad;
} eebuf;
...
void
main( void )
{
    myEEPROM_Read( 0x00,
sizeof(eebuf), &eebuf );
                        if(
strcmp(eebuf.sign,"SALV") )
{
                    strcpy(
&eebuf.sign, "SALV" );
            eebuf.psc[0] =
0xFF;
            eebuf.psc[1] =
0xFF;
            eebuf.psc[2] =
0xFF;
        eebuf.cnt_readok =
0;
        eebuf.cnt_readbad
= 0;
        eebuf.cnt_writeok
= 0;
        eebuf.cnt_writebad
= 0;
        myEEPROM_Write( 0,
sizeof(eebuf), &eebuf );
        flags.setpsc = 1;
...
}
```

una stringa dalla memoria Flash alla memoria RAM è stata utilizzata la funzione *strcp_c()*, presente nel modulo mylibs.c, di cui abbiamo parlato in precedenza.
Per chiarire questi concetti, guardiamo il **listato 1**. Le due stringhe *str_welcome1* e *str_welcome2*, così come tutte le altre stringhe utilizzate nell'applicazione, sono allocate dal compilatore nella memoria Flash, poiché dichiarate come costanti, così da risparmiare spazio nella esigua RAM. L'inizializzazione dell'LCD avviene mediante la *Lcd_Custom_Config()* e inviando il comando LCD_CURSOR_OFF per disabilitare la visualizzazione del cursore lampeggiante.
Subito dopo vengono visualizzate le due stringhe di benvenuto sulle due righe del display, utilizzando la *myLcd_Custom_Out()*. Come si può vedere sempre in **listato 1**, la *myLcd_Custom_Out()* utilizza la *strcp_c()* per copiare la stringa dalla Flash in RAM e, successivamente, la *Lcd_Custom_Out()* che visualizza questa stringa in RAM sul display.
Più avanti nel codice sarà necessario scrivere sul display un solo carattere, per

LISTATO 4
Routine principale

```
    ...
    /* CICLO PRINCIPALE */
    while( 1 ) {
        unsigned short k, event;
        event = 0;

        /* Legge l'evento. */
        if( flags.redraw ) {
            /* Ridisegna la schermata. */
            event = EVENT_DRAW;
            flags.redraw = 0;
            /* Al ridisegno della schermata, il cursore viene
nascosto. */
            Lcd_Custom_Cmd( LCD_CURSOR_OFF );
        } else if( flags.mc_inserted && !MC_INSERTED() ) {
            /* smartcard estratta. */
            event = EVENT_MCEXTRACTED;
            flags.mc_inserted = 0;
        } else if( !flags.mc_inserted && MC_INSERTED() ) {
            /* smartcard inserita. */
            event = EVENT_MCINSERTED;
            flags.mc_inserted = 1;
        } else if( (k=key_read())!=0xFF )
            /* Un tasto è stato premuto. */
            event = EVENT_KEY0 + k;

        /* Propaga l'evento alla schermata corrente. */
        if( event ) {
            switch( view ) {
                case VIEW_MAIN:
                    view_Main( event );
                        break;
                    case VIEW_SETUP:
                        view_Setup( event );
                        break;
                    case VIEW_COUNTERS:
                        view_Counters( event );
                        break;
                    case VIEW_DATA:
                        view_Data( event );
                        break;
                    case VIEW_SETPSC:
                        view_SetPSC( event );
                        break;
            }
        }
    }
```

LISTATO 5
Struttura dei dati memorizzati nella smartcard

```
struct {
  char name[17];
  char credit[7];
  char null[7];
  unsigned short chk;
} mcbuf;
```

esempio il numero appena premuto sul tastierino numerico. Per fare ciò, viene utilizzata la funzione di libreria *Lcd_Custom_Chr_Cp()* che scrive il carattere passato come argomento alla posizione corrente del cursore sul display. Per modificare la posizione del cursore utilizziamo un trucco. Per esempio, mediante l'istruzione *myLcd_Custom_Out(2, 10, "")*; non viene scritta alcuna stringa sul display, in quanto passiamo una stringa nulla, ma otteniamo il risultato di forzare la posizione del cursore alla posizione (2,10), cioè al 10° carattere della seconda riga. È proprio in questa posizione che verrà visualizzato il carattere passato come argomento alla funzione *Lcd_Custom_Chr_Cp()*.

Smartcard

C'è poco da dire a proposito della gestione della smartcard. Tutte le operazioni (reset, inserimento del PSC, modifica del PSC, lettura/scrittura della MM, ecc) vengono fatte richiamando le funzioni presenti nel modulo SLE4442.c di cui abbiamo scritto in precedenza. Qui vale la pena notare che l'inserimento/estrazione di una carta all'interno della fessura viene rilevata mediante la doppia macro:

```
#define MC_SW          PORTA.F0
#define MC_INSERTED()   (MC_SW)
```

La prima si limita a definire il pin (RA0) a cui è collegato lo switch, normalmente chiuso dello zoccolo ISO7816 della smartcard; la seconda è una macro che può essere usata nelle espressioni C laddove serva sapere se una carta sia effettivamente inserita nella fessura del connettore.

Tastierino 4x4

Come già detto nella puntata precedente, il tastierino 4x4 viene gestito mediante multiplazione delle righe: in un determinato istante si abilita una sola riga e si leggono i tasti premuti su quella riga, per passare dopo a quella successiva. Quando si ha a che fare con tasti meccanici, è necessario prendere delle precauzioni contro gli effetti del rimbalzo. Tale effetto si verifica nel momento della pressione di un tasto momentaneo (push-button) che provoca delle oscillazioni meccaniche, le quali si traducono in oscillazioni sulla tensione che misura lo stato del tasto. Queste oscillazioni scompaiono dopo un certo tempo che dipende dalle caratteristiche del tasto e possono indurre in errore il microcontrollore che potrebbe interpretarle come numerose pressioni piuttosto che una sola. Una tecnica semplice di anti-rimbalzo, che permette di evitare il rilevamento di pressioni non volute, è quella di mascherare la lettura del tasto per un certo periodo di tempo (maggiore del periodo di

LISTATO 6
Struttura della funzione view_Data()

```
void
view_Data( unsigned short event )
{
    static unsigned short idx;

    switch( event ) {

        case EVENT_DRAW:
            /* Scrivi nome sulla prima riga. */
            Lcd_Custom_Out( 1, 1, mcbuf.name );
            /* Scrivi credito sulla seconda riga. */
            myLcd_Custom_Out( 2, 1, str_credit );
            Lcd_Custom_Out( 2, 10, mcbuf.credit );
            flags.creditmod = 0;
            flags.creditmodend = 0;
            break;

        case EVENT_KEY0:
        ...
        case EVENT_KEY9:
            ...
            break;

        case EVENT_KEYE:
            if( flags.creditmod || flags.creditmodend )
                /* Annullamento dell'operazione di modifica
credito. */
                flags.redraw = 1;
        break;

        case EVENT_KEYF:
            ...
            break;

        case EVENT_MCEXTRACTED:
            LED_RED_OFF();
            LED_GREEN_OFF();
            flags.redraw = 1;
            flags.creditmod = 0;
            view = VIEW_MAIN;
            break;
    }
}
```

rimbalzo) subito dopo aver individuato una pressione. Questa tecnica è stata implementata nel nostro firmware mediante la periferica interna al microcontrollore denominata Timer1. Esso è un timer a 16 bit che incrementa di una unità il suo valore a ogni colpo di clock. Noi lo abbiamo configurato in modo che faccia "scattare" un interrupt ogni 13 ms circa.
Nel **listato 2** si trovano le parti di codice relative alla gestione del tastierino, compresa l'ISR (*Interrupt Service Routine*), cioè la routine che viene automaticamente richiamata dal microcontrollore in presenza di un interrupt da servire. Nella funzione principale *main()* la periferica Timer1 viene configurata in modalità timer, cioè il registro TMR1 a 16 bit viene incrementato automaticamente a ogni colpo di clock. Inoltre viene abilitato anche l'interrupt relativo all'overflow del registro TMR1, che si verifica quando il registro passa dal valore *0xFFFF a 0x0000*. Ogni quanto tempo scatta l'interrupt? Il calcolo è abbastanza semplice se consideriamo che ci vogliono 65536 colpi di clock per arrivare all'overflow di un registro a 16 bit e che un colpo di clock dura Fosc/4 (dove Fosc è la frequenza del quarzo esterno, nel nostro caso 20 MHz):

```
65536 * 4 / 20000000 ~= 13ms
```

A ogni chiamata dell'ISR, quindi ogni 13 ms circa, la variabile *port* indica la pressione dei quattro tasti della riga selezionata. La complessità derivante da una prima lettura dell'istruzione:

```
port = ~((PORTB>>4)|0xF0);
```

può essere allontanata notando che le colonne sono collegate ai pin RB4-RB7 (questo spiega lo shift a destra di quattro posizioni e l'OR con la costante esadecimale *F0*) e che un tasto premuto provoca un livello basso sul pin corrispondente (e questo spiega la negazione). Per esempio, se dopo l'esecuzione di questa istruzione, la variabile *port* è pari a *0x06*, i tasti collegati su RB5 e RB6 risultano premuti. Mediante la variabile *tmr_isr*, il controllo sui tasti premuti viene ritardato per annullare l'effetto del rimbalzo. Normalmente tale variabile è nulla e quindi viene eseguito il controllo dei tasti premuti a ogni interrupt. Non appena un tasto viene premuto, esso viene salvato nella variabile *key* e la variabile *tmr_isr* viene posta a 20, in modo da ritardare la successiva lettura dei tasti dopo che le oscillazioni del rimbalzo si esauriscono.
Il programma principale si limiterà a leggere la variabile *key*: se essa è pari a 0xFF, nessun tasto risulta premuto, altrimenti essa conterrà il valore del tasto premuto. Tale valore viene ottenuto leggendo semplicemente l'array a doppio indice *keytable[ ][ ]* che indica la disposizione dei tasti sul tastierino, la loro interconnessione in righe e colonne e con i pin del microcontrollore. Se utilizzate un diverso tastierino è molto probabile che dobbiate modificare questo array.
Le variabili *row* e *rowmask* rappresentano la riga selezionata: la prima indica il numero di riga (a partire da 0) e viene usata come primo indice dell'*array keytable*; la seconda indica il bit associato alla riga e viene utilizzato per attivare il pin corrispondente.
L'ultima istruzione dell'ISR è importante, perché azzera il flag dell'interrupt relativo alla periferica Timer1, comunican-

LISTATO 7
Gestione della pressione dei tasti numerici e del tasto F in view_Data()

```
    case EVENT_KEY0:
    ...
    case EVENT_KEY9:
        if( flags.creditmod ) {
            /* È stato premuto un tasto numerico. */
            strbuf[idx] = '0' + event-EVENT_KEY0;
            Lcd_Custom_Chr_Cp( strbuf[idx] );
            /* Passa alla cifra successiva. */
            if( ++idx==3 )
                ++idx;        /* Salta la virgola. */
            /* Sposta il cursore. */
            Lcd_Custom_Out( 2, 10+idx, "" );
            if( idx==6 ) {
                flags.creditmod = 0;
                flags.creditmodend = 1;
            }
        }
        break;

    case EVENT_KEYF:
        if( flags.creditmodend ) {
            /* Conferma del nuovo credito appena scritto. */
            memcpy( mcbuf.credit, strbuf, sizeof(mcbuf.cred-
it) );
            set_credit();
            flags.creditmodend = 0;
            flags.redraw = 1;
        } else if( !flags.creditmod ) {
            /* L'utente vuole modificare il credito. */
            flags.creditmod = 1;
            flags.creditmodend = 0;
            idx = 0;
            memcpy( strbuf, mcbuf.credit, sizeof(mcbuf.cred-
it) );
            /* Visualizza il cursore lampeggiante sulla pri-
ma cifra. */
            Lcd_Custom_Out( 2, 10, "" );
            Lcd_Custom_Cmd(LCD_BLINK_CURSOR_ON);
            LED_GREEN_OFF();
            LED_RED_OFF();
        }
        break;
```

do alla logica del microcontrollore che l'evento è stato servito e l'esecuzione del "normale" programma può continuare (senza questa istruzione, si entrerebbe continuamente nell'ISR).

EEPROM

Il PIC ha una EEPROM interna che abbiamo utilizzato per memorizzare alcune impostazioni non volatili, in modo che non vengano "perse" dopo uno spegnimento. All'accensione, durante la fase di inizializzazione, queste impostazioni vengono lette dalla EEPROM, così come mostrato nel **listato 3**.

Tutti i dati memorizzati in EEPROM sono organizzati in una struct dal nome *eebuf*. All'interno troviamo: una firma, il PSC da utilizzare per le operazioni di scrittura nella smartcard e alcuni contatori a fini statistici. Vale la pena spendere alcune parole riguardo la firma (membro sign della struttura). Subito dopo la programmazione del microcontrollore, la EEPROM interna è al suo valore di default, cioè tutti i byte sono 0xFF, quindi non è affidabile. In queste condizioni è necessario scrivere dei valori di default. La firma viene utilizzata proprio per distinguere una EEPROM vergine (probabilmente subito dopo la programmazione del microcontrollore) o corrotta da una già inizializzata. Abbiamo utilizzato come firma la sequenza di quattro caratteri (più il carattere nullo che identifica la fine della stringa in C): "SALV". Subito dopo aver letto i parametri dalla EEPROM, viene controllato che la firma coincida con "SALV": se è così, i dati appena letti possono essere ritenuti corretti, altrimenti vengono scritti dei valori di default (PSC=*0xFFFFFF* e i contatori tutti azzerati). Alla fine di questa operazione, viene settato il flag setpsc che indica che è necessario chiedere subito all'utente il valore del PSC da utilizzare, visto che non ne conosciamo uno attendibile (è grazie a questo accorgimento che, alla prima accensione del lettore, viene visualizzata la schermata per l'impostazione del PSC).

Routine principale

Il ciclo principale del programma è racchiuso all'interno di un ciclo infinito *while(1)* ed è mostrato nel **listato 4**.
Esso è stato organizzato a eventi: scrittura del display, inserimento o estrazione della smartcard, pressione di un tasto. A seconda della schermata in cui ci troviamo (variabile *view*), viene richiamata una funzione diversa, passando come parametro l'evento. Per esempio, trovandoci nella schermata principale (*view=VIEW_MAIN*) e premendo il tasto 3, verrà richiamata la funzione *view_Main()*, passando come parametro il valore EVENT_KEY3. Come si può vedere, questa è una tecnica molto efficace per gestire diverse schermate, ognuna delle quali ha un compito diverso. La complessità di ogni singola schermata è ben separata dalle altre ed è più facilmente manutenibile.
A questo punto dovremmo descrivere il funzionamento delle funzioni relative alle singole schermate: *view_Main()* che gestisce la schermata principale; *view_Setup()* che gestisce la schermata di setup; *view_Counters()* che gestisce la schermata dei contatori; *view_Data()* che gestisce la schermata dei dati memorizzati nella smartcard; *view_SetPSC()* che gestisce la schermata per modificare il PSC. Per evitare di appesantire troppo la discussione, ci limitiamo a illustrare il codice relativo alla schermata dei dati.

LISTATO **8**
Funzione set_credit()

```
void
set_credit( void )
{
    /* Lettura dell'Error Counter. */
    if( mc_readEC()!=0x07 ) {
        LCD_ERRMESSAGE( str_cardblocked1, str_cardblocked2 );
        return;
    }

    /* Azzera un bit dell'Error Counter. */
    mc_writes( 00, 0x03 );
    /* Confronta i 3 byte del PSC. */
    mc_compare( 01, eebuf.psc[0] );
    mc_compare( 02, eebuf.psc[1] );
    mc_compare( 03, eebuf.psc[2] );
    mc_writes( 00, 07 );    /* Ricarica l'Error Counter. */
    if( mc_readEC()!=0x07 ) {
        /* Il PSC molto probabilmente è errato. */
        LCD_ERRMESSAGE( str_updateerr1, str_updateerr2 );
        set_counter( COUNTER_WRITEBAD, eebuf.cnt_writebad+1 );
        return;
    }

    /* Calcola la nuova checksum. */
    mcbuf.chk = checksum( &mcbuf, sizeof(mcbuf)-1 );
    /* Scrive il credito. */
      mc_writem( MCADDR_CREDIT, sizeof(mcbuf.credit), &(m-
cbuf.credit) );
    /* Scrive la checksum. */
    mc_writem( MCADDR_CHK, sizeof(mcbuf.chk), &(mcbuf.chk) );

    /* Scrittura plausibilmente andata a buon fine. Per mag-
giore sicurezza,
     * rileggiamo il contenuto della smartcard. */
    if( read_data(&tmpbuf)!=OK )  {
        /* Checksum errata, scrittura non andata a buon fine.
*/
        LCD_ERRMESSAGE( str_updateerr1, str_updateerr2 );
        set_counter( COUNTER_WRITEBAD, eebuf.cnt_writebad+1 );
        return;
    }
    if( memcmp(&tmpbuf,&mcbuf,sizeof(mcbuf)) ) {
        /* Il buffer letto non corrisponde con quello che v-
olevamo scrivere.
         * Scrittura errata. */
        LCD_ERRMESSAGE( str_updateerr1, str_updateerr2 );
        set_counter( COUNTER_WRITEBAD, eebuf.cnt_writebad+1 );
        return;
    }

    /* Se siamo qui, la scrittura è avvenuta correttamente. */
    set_counter( COUNTER_WRITEOK, eebuf.cnt_writeok+1 );
    LED_GREEN_ON();
```

view_Data()

Dalla schermata principale, visualizzata subito dopo l'accensione, si passa alla schermata dei dati non appena viene inserita una smartcard valida nel lettore. Per verificare la validità della smartcard inserita, vengono eseguiti alcuni controlli sull'ATR (che deve corrispondere all'ATR di una SLE4442) e sui dati memorizzati nella Main Memory della smartcard. Si veda il **listato 5** per la struttura utilizzata per memorizzare i dati nella smartcard. Molto semplicemente, si tratta di una stringa di 17 caratteri per il nome (16 più il carattere nullo di fine stringa) e di 7 caratteri per il credito. Il credito permette di memorizzare un numero con tre cifre per la parte intera, una virgola e due cifre per la parte decimale, oltre al solito terminatore nullo. Ci sono 7 byte non utilizzati e infine una checksum, cioè un byte che è sempre pari allo XOR di tutti i byte precedenti della struttura. In tutto sono 32 byte.
All'inserimento della smartcard, la routine *view_Main()* viene richiamata con l'evento *EVENT_MCINSERTED*. Dopo il controllo dell'ATR, vengono letti i dati dalla smartcard e controllata la checksum. Se questa è valida, si passa alla schermata dei dati impostando la variabile *view* al valore *VIEW_DATA*, in modo che tutti gli eventi successivi vengano gestiti proprio dalla funzione *view_Data()*, la cui struttura è mostrata nel **listato 6**. L'evento *EVENT_DRAW* viene eseguito una volta quando si passa nella nuova schermata. Qui ci limitiamo a scrivere i dati sul display (nome e credito memorizzato nella smartcard) e ad azzerare due flag utilizzati successivamente.
L'evento EVENT_MCEXTRACTED viene eseguito quando la smartcard è estratta dal lettore. In questo caso ci limitiamo a spegnere i LED eventualmente accesi e a tornare alla schermata principale *VIEW_MAIN*. Si noti l'attivazione del flag *redraw* che andrà a forzare il disegno della nuova schermata. In questa schermata è possibile premere il tasto F per modificare il credito memorizzato nella smartcard e il tasto E per annullare l'operazione di modifica. L'evento riguardate la pressione del tasto E viene gestito quando siamo durante la modifica del credito (flag *creditmod* attivo), oppure quando la modifica del credito è terminata e il lettore si aspetta la conferma (flag

creditmodend attivo). In entrambi i casi, viene attivato il flag *redraw* che andrà a ridisegnare tutta la schermata, annullando l'operazione di modifica.
Analizziamo, ora, la gestione dei due eventi più complicati: la pressione di uno dei tasti numerici e la pressione del tasto F (si veda il **listato 7**). La pressione dei tasti numerici è gestita soltanto quando siamo nella fase di modifica del credito (flag *creditmod* attivo), altrimenti tali eventi vengono ignorati. Il tasto numerico premuto viene convertito in carattere dall'istruzione:

```
strbuf[idx] = '0' + event-
EVENT_KEY0;
```

e memorizzato nell'array *strbuf[]* alla posizione *idx* (variabile dichiarata statica all'interno di *view_Data()* in modo che non perda il suo valore ritornando nella funzione). Questo carattere viene successivamente visualizzato a display e la variabile *idx* viene incrementata per ospitare il successivo carattere, stando attenti a saltare la virgola (in posizione 3), che non è modificabile. Infine viene spostato il cursore sulla posizione della successiva cifra da modificare e si controlla se è stato modificato l'ultimo numero (siamo arrivati alla posizione 6). In caso affermativo, viene azzerato il flag di modifica del credito e impostato il flag di modifica del credito terminata.
Studiamo ora l'evento relativo alla pressione del tasto F che si può dividere in due parti: la pressione del tasto F, quando l'utente ha appena terminato la digitazione del nuovo credito (primo ramo dell'if), e la pressione del tasto F, quando non siamo ancora entrati nella modalità di modifica del credito (subito dopo l'inserimento della smartcard).
Il primo caso è relativo alla conferma del nuovo valore appena digitato: viene modificato il credito in RAM (struttura *mcbuf*), copiando il nuovo valore dalla stringa *strbuf[ ]* ed eseguita la funzione *set_credit()* che si preoccuperà di scrivere questo nuovo valore nella smartcard.
Il secondo caso riguarda l'attivazione della modalità di modifica del credito visualizzato. Oltre a impostare correttamente i due flag di cui abbiamo ampiamente parlato e a inizializzare la variabile *idx*, il credito memorizzato nella smartcard viene copiato nell'array *strbuf[ ]* che rappresenterà il nuovo valore (non possiamo usare *mcbuf*.credit poiché l'utente potrebbe annullare l'operazione). Viene inoltre impostata la posizione del cursore in modo che corrisponda alla prima cifra del credito e attivata la visualizzazione grafica del cursore lampeggiante a display. Infine, vengono spenti i due LED.
La relativa semplicità della funzione *view_Data()* non deve ingannare: tutto il "lavoro sporco" della scrittura del nuovo credito nella smartcard è effettuato dalla funzione *set_credit()*, mostrata nel **listato 8**. Per modificare il credito memorizzato nella smartcard è necessario scrivere nella Main Memory, quindi è necessario inserire il PSC corretto. Per farlo, viene subito controllato che l'Error Counter sia carico, cioè si abbiano ancora "tre gettoni" disponibili (avremmo potuto tentare l'inserimento del PSC anche con un solo "gettone" disponibile, ma, in caso di ulteriore fallimento, avremmo ottenuto una carta completamente bloccata). Successivamente viene "speso un gettone", inserito il PSC (memorizzato nella EEPROM del microcontrollore) e ricaricato l'Error Counter. A questo punto è possibile stabilire se l'operazione di inserimento del PSC è andata a buon fine, verificando che l'Error Counter sia nuovamente carico. Dopo aver calcolato la nuova checksum, viene scritto il nuovo credito e la nuova checksum nella Main Memory della smartcard. Per ulteriore conferma, andiamo a leggere i dati dalla smartcard, controllando nuovamente la correttezza della checksum e confrontando i dati appena letti con quelli che volevamo scrivere. Se tutto è OK, viene acceso il LED verde. Se la funzione viene interrotta per un errore, viene acceso il LED rosso. Si noti la chiamata della funzione *set_counter()* che va a memorizzare, nella EEPROM del microcontrollore, un nuovo valore di uno dei quattro contatori, individuati simbolicamente da *COUNTER_READOK, COUNTER_READBAD, COUNTER_WRITEOK, COUNTER_WRITEBAD.*

CONCLUSIONI

Il lettore di smartcard stand-alone descritto in questi due articoli può essere realizzato da chiunque in quanto i componenti utilizzati sono molto comuni e le informazioni necessarie sono tutte presenti negli articoli e sulle pagine web relative agli articoli. Oltre a risolvere un problema contingente relativo all'applicazione di fidelizzazione, nella descrizione, soprattutto del codice sorgente, abbiamo cercato di mostrare le linee guida generali per poter implementare un firmware complesso che deve eseguire numerosi compiti (lettura/scrittura dei dati dalla smartcard, gestione di schermate su display, lettura di una tastiera, ecc).
I più intraprendenti potranno divertirsi, oltre a realizzare lo stesso prototipo che abbiamo realizzato noi, a modificare il codice sorgente per personalizzarlo e/o ampliarne le funzionalità. Una proposta interessante può essere quella di far inizializzare una smartcard vergine direttamente dal nostro lettore stand-alone piuttosto che tramite il software su PC e l'UniReader. Per fare ciò, è necessario creare una nuova schermata di inizializzazione della smartcard dove sarà necessario modificare il PSC (e quindi sarà necessario conoscere il PSC della carta vergine, magari chiedendolo direttamente all'utente) e scrivere dei dati di default. La parte più complicata è la scrittura del nome alfanumerico che deve avvenire tramite un tastierino esadecimale numerico. Una possibile soluzione è quella di utilizzare il tastierino alla maniera dei telefonini: ogni tasto numerico rappresenta, oltre alla cifra, anche 3 o 4 lettere alfabetiche. La pressione ripetuta a tempi stretti del tasto permette di scegliere la lettera o la cifra, prima di passare alla posizione successiva. ❑

PER *approfondire...*

Sulla pagine web relative agli articoli trovere i datasheet dei principali componenti utilizzati nel progetto (il chip della smartcard SLE4442, il microcontrollore PIC 16F876A).
La lettura del tutorial sulle smartcard già pubblicato su Fare Elettronica n. 227 e n. 234 (maggio e dicembre 2004) è di vitale importanza per conoscere il mondo delle smartcard, le SLE4442 in particolare e il lettore UniReader.

CODICE MIP 2757067

Con il patrocinio del: Ministero dello Sviluppo Economico

Expo Elettronica VICENZA

CODICE MIP 2778651

6-7 febbraio 2010

FIERA DI VICENZA (Pad. L)

ore 9/18

MOSTRA MERCATO DI:

informatica,
telefonia mobile e fissa,
hardware e software,
ricezione satellitare,
hobbistica, DVD e Games,
materiali di consumo.

MERCATINO DELL'USATO

rif. Fare Elettronica

vale come ridotto

Fotografa questo codice e scarica il ticket ridotto direttamente sul tuo smartphone

SNAP IT

Organizzazione e informazioni:
BLU Nautilus tel 0541 439573
www.expoelettronica.it

I Resistori ELETTRONICI

I resistori, chiamati spesso e impropriamente resistenze, sono componenti fondamentali nella realizzazione di qualunque circuito elettronico. Sono prodotti in un'ampia varietà di forme, di valori ohmici e di potenze. Vediamo quali sono gli aspetti tecnologici che li caratterizzano e li diversificano e quali sono le codifiche impiegate per identificarli tecnicamente

I resistori reali, chiamati spesso e impropriamente resistenze (più propriamente il resistore è l'oggetto fisico, mentre la resistenza è la grandezza fisica che ne quantifica il valore ohmico), sono caratterizzati dal valore della loro resistenza elettrica espressa in ohm (simbolo Ω) e dalla massima potenza (energia per unità di tempo) espressa in watt che sono in grado di dissipare senza distruggersi o deteriorarsi per surriscaldamento. Sebbene esistano moltissimi tipi di resistori, in questo articolo ci occuperemo prevalentemente di quelli che vengono impiegati in campo elettronico.

ASPETTI FISICI E TECNOLOGICI

La resistenza di un filo conduttore è data dalla seguente semplice relazione:

$$R = \rho \frac{l}{S}$$

dove l è la sua lunghezza, S la sezione e ρ la resistività del materiale. Essa vale alla temperatura alla quale è riferita la resistenza specifica ρ (resistività); quest'ultima è infatti funzione della temperatura. Tale dipendenza è valutabile mediante il coefficiente di temperatura lineare α e può essere descritta attraverso la seguente relazione linearizzata:

$$\rho(T_2) = \rho(T_1)\left[1 + \alpha_{T1}(T_2 - T_1)\right]$$

dove α_{T1} è il coefficiente di temperatura alla temperatura T_1 (la relazione può essere considerata valida in un intorno abbastanza ampio di temperature intorno a T_1). Queste semplici relazioni sono sufficienti per definire quali siano i parametri fisici che incidono sul valore ohmico offerto ai suoi capi da un resistore. Ciò è comprensibile dall'analisi della **figura 1** in cui è riportata la struttura fisica rispettivamente di una resistenza bobinata (in cui l'elemento resistivo è costituito da un filo avvolto intorno a un cilindro ceramico), di una resistenza a strato metallico (in cui l'elemento resistivo è un'elica metallica ancora una volta avvolta intorno a un nucleo cilindrico ceramico) e di una resistenza a strato di carbone (in cui vi è un'elica di carbone che si sviluppa intorno al solito corpo cilindrico ceramico). A parte il caso della resistenza bobinata quindi, l'elemento resistivo è realizzato depositando sulla superficie del cilindro uno strato di lega metallica, ossidi metallici o carbone appunto, successivamente inciso opportunamente al laser, con andamento elicoidale, in modo da ottenere il valore ohmico desiderato.
Il corpo della resistenza così ottenuto è rivestito di uno strato isolante simile a una ver-

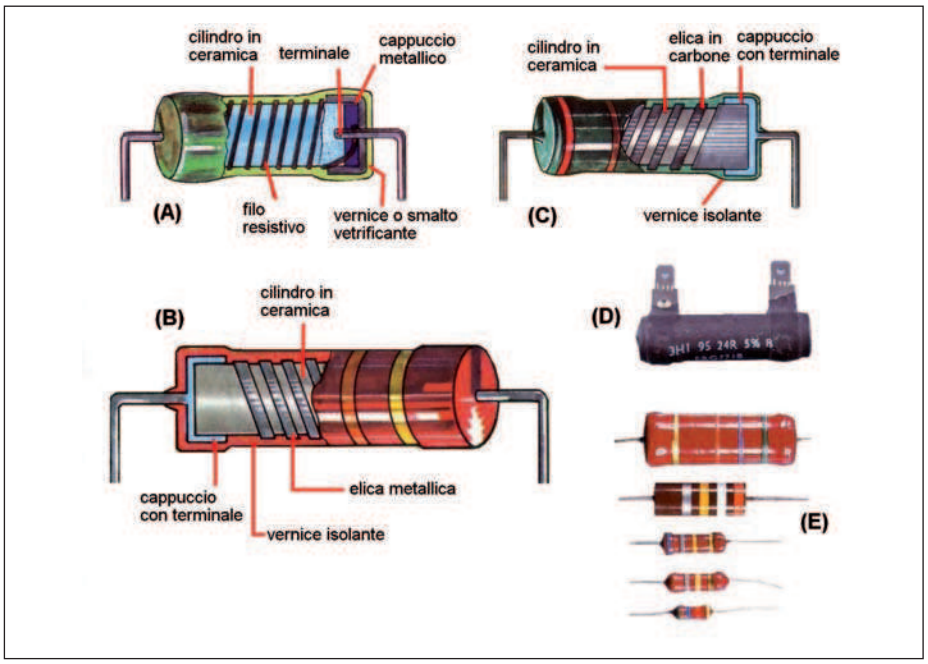


Figura 1: struttura fisica di una resistenza bobinata (A), di una resistenza a strato metallico (B), a strato di carbone (C) e aspetto esterno di alcune resistenze reali (E) [1].

nice su cui sono serigrafate le fasce colorate che ne codificano il valore ohmico nominale (codice dei colori). In commercio esistono resistenze di questo tipo con dimensioni tali da rispondere, in termini di potenza, allo standard 1/8W, 1/4W, 1/3W, 1/2W, 1W e 2W, con tolleranze sul valore ohmico tipicamente pari a 1%, 2%, 5%, 10% e 20%.
Le resistenze bobinate, diversamente dalle altre, sono in realtà utilizzate per potenze più elevate di 2W. Le resistenze a strato metallico sono invece impiegate per ottenere valori ohmici di elevata precisione: al posto dello strato di carbone vi è lo strato di metallo lavorato a spirale. La struttura fisica descritta conferisce in realtà al resistore una componente induttiva, ininfluente in applicazioni a bassa frequenza, ma di disturbo a frequenze elevate. Per queste applicazioni si preferiscono resistori antinduttivi, realizzati comprimendo una miscela di polveri composite e resine, a formare un cilindro con dimensioni simili ai resistori a strato oppure i resistori di tipo SMD. È importante sottolineare come per essere stabile nel suo valore ohmico, il componente non debba surriscaldarsi; inoltre il valore ohmico del resistore può essere più o meno insensibile alla variazione della temperatura anche in funzione del materiale di cui è fatto.

Un parametro fondamentale: la potenza

Le resistenze sono caratterizzate non solo dal valore ohmico espresso in ohm ma anche da un valore di potenza espressa in watt. Quando si disegna un circuito elettrico si indica vicino a ogni resistenza il relativo valore ohmico, tuttavia nel momento in cui si realizza fisicamente il circuito non è sufficiente conoscere il valore ohmico della resistenza. Infatti, una cosa è impiegarla perché sia attraversata da una corrente pari a 100 mA, cosa ben diversa è impiegarla per una corrente pari a 10 A. Oltre al valore ohmico, il parametro fondamentale che determina il campo di impiego del resistore è quindi la potenza massima (espressa in watt) che esso è in grado di dissipare senza che si riscaldi a tal punto da distruggersi o danneggiarsi. Le resistenze elettroniche più piccoli in termini di potenza (1/8, 1/4,1/2 W) hanno la tipica forma di piccoli cilindri provvisti di sottili reofori terminali (**figura 2**) e sono realizzate generalmente in grafite. Resistenze analoghe solo leggermente più grandi sono realizzate per potenze pari a 1 o 2W. Resistenze di potenza ancora più elevata (potenze tipicamente pari a 3-5-7-10-15-20-30 W) impiegano invece nichelcromo e presentano un tipico corpo rettangolare in ceramica (**figura 3** e **figura 5**). Mentre per le prime il valore ohmico è riportato sul corpo del componente attraverso una speciale codifica che prende il nome di "codice dei colori", per le ultime, il valore ohmico, insieme al valore di potenza massima è generalmente stampato sul corpo in forma numerica esplicita. Le dimensioni della resistenza sono in ogni caso fisicamente proporzionali alla potenza dissipabile (le dimensioni fisiche del resistore tipicamente crescono al crescere della potenza, ma non al crescere del valore ohmico). In **tabella 1** sono riportate le dimensioni tipiche delle resistenze ohmiche impiegate in elettronica per potenze pari rispettivamente a 1/8W, 1/4W, 1W e 2W.
Fondamentalmente, da un punto di vista di tolleranza, ci sono due tipi di resistenze a basso wattaggio per circuiti elettronici: resistenze a precisione standard e resistenze comunemente dette di precisione. La differenza sta nella diversa tolleranza del valore nominale; per il primo tipo tale valore può essere pari a ±5%, ±10% e ±20%, mentre per il secondo tipo il valore è inferiore o uguale a ±2%. Normalmente si adoperando le normalissime resistenze a precisione standard e solo in casi specifici, come nei circuiti di misura, si impiegano resistenze di precisione. Per esempio, una resistenza con valore nominale pari a 82000 Ω e tolleranza del 5% presenta un valore reale che può variare tra 82000*0,95 = 77900 Ω e 82000*1,05 = 86100 Ω. Al contrario la stessa resistenza con tolleranza dell'1% presenta un valore reale che può variare tra 82000*0,99 = 81180 Ω e 82000*1,01 = 82820 Ω.

I valori ohmici standard dei resistori elettronici

I valori ohmici nominali dei resistori sono divisi in sei serie standard, ciascuna caratterizzata da una precisa tolleranza (**tabella 2**). Le serie utilizzate più comunemente sono la E12 e la E24 (tolleranze pari rispettivamente a ±10% e ±5%). Ormai sono invece raramente utilizzate le resistenze della serie E6 (tolleranza ±20%) che stanno sostanzialmente scomparendo, mentre le resistenze delle serie E48, E96 ed E192, avendo una tolleranza inferiore al 2%, sono impiegate in applicazioni di precisione (e risultano ovviamente più costose). Quando si dimensiona un circuito elettronico si eseguono dei calcoli che portano a definire i valori e le caratteristiche tecniche di ciascun componente e tra queste quelle dei resistori. I valori risultanti dai calcoli, tuttavia, non sono mai quelli effettivamente impiegati nel circuito fisico; infatti in commercio esistono solo resistenze di valore standard suddivise per tolleranza nelle serie precedentemente elencate. Al dimensionamento teorico di un circuito segue, pertanto, in fase di realizzazione,

Figura 2: resistori da 1/8 di watt a ossido di metallo con terminali assiali a saldare [2].

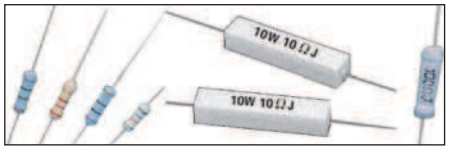
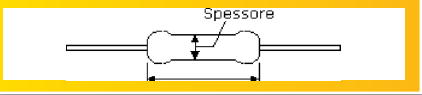


Figura 3: le resistenze da 1/8-1/4-1/2-1 watt hanno forma cilindrica. Il valore ohmico si ricava dalle fasce colorate incise sul corpo. Le resistenze da 3-5-7-10-15 watt hanno generalmente forma rettangolare in ceramica con sopra stampato il valore ohmico e la potenza in watt.

Tabella 1 - Relazione Dimensioni-Potenza per le resistenze elettroniche da 1/8W a 2W

POTENZA MAX (W)	SPESSORE (mm)	LUNGHEZZA (mm)
1/8	2	3
1/4	2	6
1	3.5	12
2	5	15



1,0 ohm	10 ohm	100 ohm	1.000 ohm	10.000 ohm	100.000 ohm	1,0 megaohm
1,2 ohm	12 ohm	120 ohm	1.200 ohm	12.000 ohm	120.000 ohm	1,2 megaohm
1,5 ohm	15 ohm	150 ohm	1.500 ohm	15.000 ohm	150.000 ohm	1,5 megaohm
1,8 ohm	18 ohm	180 ohm	1.800 ohm	18.000 ohm	180.000 ohm	1,8 megaohm
2,2 ohm	22 ohm	220 ohm	2.200 ohm	22.000 ohm	220.000 ohm	2,2 megaohm
2,7 ohm	27 ohm	270 ohm	2.700 ohm	27.000 ohm	270.000 ohm	2,7 megaohm
3,3 ohm	33 ohm	330 ohm	3.300 ohm	33.000 ohm	330.000 ohm	3,3 megaohm
3,9 ohm	39 ohm	390 ohm	3.900 ohm	39.000 ohm	390.000 ohm	3,9 megaohm
4,7 ohm	47 ohm	470 ohm	4.700 ohm	47.000 ohm	470.000 ohm	4,7 megaohm
5,6 ohm	56 ohm	560 ohm	5.600 ohm	56.000 ohm	560.000 ohm	5,6 megaohm
6,8 ohm	68 ohm	680 ohm	6.800 ohm	68.000 ohm	680.000 ohm	6,8 megaohm
8,2 ohm	82 ohm	820 ohm	8.200 ohm	82.000 ohm	820.000 ohm	8,2 megaohm

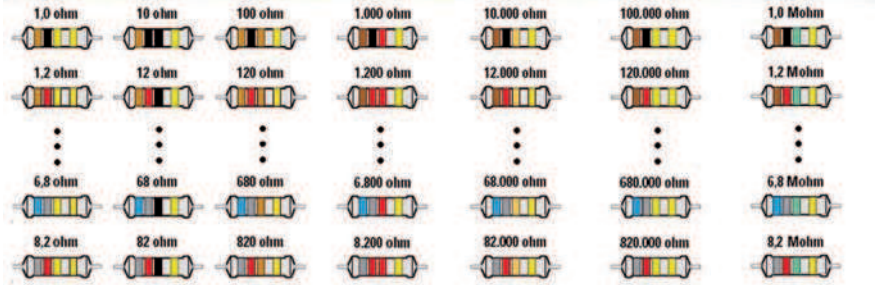


Figura 4: valori nominali della sequenza di resistenze standard E12 (tolleranza 10%).

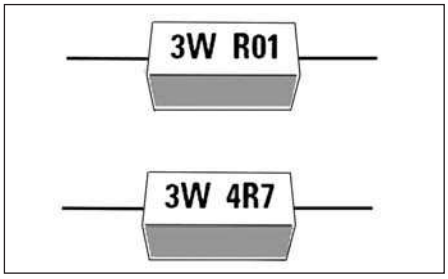


Figura 5: esempi di resistenze di potenza e relativa codifica del valore ohmico.

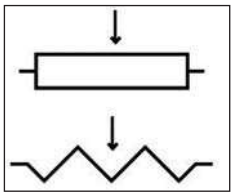


Figura 6: simbolo di un resistore variabile (trimmer).

l'impiego delle resistenze standard che presentano i valori ohmici più prossimi a quelli derivanti dai calcoli; in alternativa possono essere impiegati serie o paralleli di resistenze standard.

La normativa internazionale IEC 60063 definisce i valori standardizzati delle resistenze; il numero di valori disponibili aumentano con la precisione. Ad esempio, per la serie E6 (tolleranza 20%) sono consentiti sei valori: 10, 15, 22, 33, 47, 68 e relativi multipli di 10 (ad esempio 0,47 Ω, 4,7 Ω, 47 Ω, 470 Ω, 4,7 kΩ, 47 kΩ, 470 kΩ e così via). In funzione della tolleranza, la norma IEC 60063 definisce i valori standard riportati in **tabella 3**. In parentesi è riportata la tolleranza specifica di ciascuna serie. La serie E192 diversamente dalle altre è usata per resistori con diverse tolleranze (0,5%, 0,25% e 0,1%). I valori delle singole serie sono ottenuti dalla seguente progressione geometrica:

$$ar^0, ar^1, ar^2, ar^3, \ldots$$

Dove r ≠ 0 è la ragione e il fattore a è pari a 1. La ragione e il termine N_{esimo} risultano (n = 0 … (k-1)) rispettivamente:

$$r = \sqrt[k]{10}$$

$$N = (\sqrt[k]{10})^n$$

Il valore k varia a seconda della serie: per la E6 (20%) k=6, per la E12 (10%) k=12, per E24 (5%) k= 24, per E48 (2%) k=48, per E96 (1%) k=96 mentre per E192 (0,5%, 0,25% e 0,1%) k=192. Questo spiega anche il motivo del numero che segue la lettera E nella definizione dei valori nominali standard. La serie E6, per esempio, è caratterizzata da k=6 e quindi da 6 possibili valori numerici (le resistenze che ne fanno parte presentano questi valori ohmici e i rispettivi multipli), la serie E12 da 12 possibili valori e così via (**tabella 3**).

Tabella 2 - Classificazione delle resistenze in funzione delle tolleranze [3], [4]

E6	E12	E24	E48	E96	E192
±20%	±10%	±5%	±2%	±1%	±0.5%

Codifica delle resistenze elettroniche: il codice dei colori

Il valore ohmico delle resistenze più comunemente impiegate in elettronica non è stampato esplicitamente sul corpo del componente, bensì è serigrafato su di esso mediante una serie di fasce colorate che hanno un preciso significato (codice dei colori). Sul corpo del resistore sono riportate 4 o 5 bande colorate che lette in un determinato modo ne dichiarano il valore nominale. Le tabelle con i codici dei colori sono due, in quanto le normali resistenza hanno 4 anelli colorati, mentre le resistenze di precisione ne hanno 5 o 6. Per comprendere tale codifica è necessario far riferimento, rispettivamente, alla codifica dei Valori Numerici della 1ª e 2ª banda (anche 3ª nel caso di resistenze di precisione; **tabella 4**), alla Codifica del Moltiplicatore (3ª banda nelle resistenze tradizionali, 4ª in quelle di precisione; **tabella 5**), a quella della Tolleranza (4a banda nelle resistenze tradizionali, 5ª in quelle di precisione; **tabella 6**) ed eventualmente a quella del Coefficiente di temperatura (**tabella 7**) alla cui analisi dettagliata si rimanda. I codici a colori per i resistori fissi sono definiti dalla EIA (Tabella codici colori EIA-RS-279).

Codifica a quattro bande

Soltanto le resistenze più economiche (tolleranza 20%) hanno tre bande di colore; le altre hanno sempre una quarta banda in colore argento o oro, che indica il valore della tolleranza: argento → +/- 10%; oro → +/- 5%. Poiché questi colori sono sempre presenti nell'ultima fascia, è ovvio che per leggere il valore di una resistenza si inizierà sempre dal lato opposto all'oro o all'argento. Vediamo ora la corrispondenza colore-cifra:

- prime due fasce: la regola fondamentale su cui si basa il codice dei colori è quella di far corrispondere a ciascuna cifra, che compone il valore di una resistenza, un preciso colore (**tabella 4**). La prima fascia colorata individua quindi la prima cifra del valore numerico: se questa fascia è di colore marrone la prima cifra è 1, se è rossa la cifra è a 2 e così via fino alla cifra 9 se il colore è il bianco. Cosa analoga può essere ripetuta per la seconda cifra;

FORTRONIC
electronics forum

enjoy
the electronic community

the Electronics Forum

- Opto & Display
- Power
- RF & Wireless
- Rework & Repair

2010
Tunisia
Turkey
Spain

share with us

International partner

Latium International Electronics Suppliers

with the support of

REGIONE LAZIO

SVILUPPOLAZIO

organizzazione
Tecnoimprese Scarl
Via Console Flaminio 19 • 20134 Milano
Ph +39 02 210 111 1 • Fax +39 02 210 111 222

assodel

www.fortronic.it

CODICE MIP 2767325

Tabella 3 - Valori nominali standard delle resistenze secondo la normativa internazionale IEC 60063 [3]

E6 (20%)	10	15	22	33	47	68	.	.	.	.	.	.
E12 (10%)	10	12	15	18	22	27	33	39	47	56	68	82
E24 (5%)	10	11	12	13	15	16	18	20	22	24	27	30
"	33	36	39	43	47	51	56	62	68	75	82	91
E48 (2%)	100	105	110	115	121	127	133	140	147	154	162	169
"	178	187	196	205	215	226	237	249	261	274	287	301
"	316	332	348	365	383	402	422	442	464	487	511	536
"	562	590	619	649	681	715	750	787	825	866	909	953
E96 (1%)	100	102	105	107	110	113	115	118	121	124	127	130
"	133	137	140	143	147	150	154	158	162	165	169	174
"	178	182	187	191	196	200	205	210	215	221	226	232
"	237	243	249	255	261	267	274	280	287	294	301	309
"	316	324	332	340	348	357	365	374	383	392	402	412
"	422	432	442	453	464	475	487	499	511	523	536	549
"	562	576	590	604	619	634	649	665	681	698	715	732
"	750	768	787	806	825	845	866	887	909	931	953	976
E192 (0,5%,0,25%,0,1%)	100	101	102	104	105	106	107	109	110	111	113	114
	115	117	118	120	121	123	124	126	127	129	130	132
	133	135	137	138	140	142	143	145	147	149	150	152
	154	156	158	160	162	164	165	167	169	172	174	176
	178	180	182	184	187	189	191	193	196	198	200	203
	205	208	210	213	215	218	221	223	226	229	232	234
	237	240	243	246	249	252	255	258	261	264	267	271
	274	277	280	284	287	291	294	298	301	305	309	312
	316	320	324	328	332	336	340	344	348	352	357	361
	365	370	374	379	383	388	392	397	402	407	412	417
	422	427	432	437	442	448	453	459	464	470	475	481
	487	493	499	505	511	517	523	530	536	542	549	556
	562	569	576	583	590	597	604	612	619	626	634	642
	649	657	665	673	681	690	698	706	715	723	732	741
	750	759	768	777	787	796	806	816	825	835	845	856
	866	876	887	898	909	919	931	942	953	965	976	988

Tabella 4 – Codifica dei Valori Numerici 1ª e 2ª banda (anche 3ª nel caso di resistenze di precisione)

Nero	0
Marrone	1
Rosso	2
Arancio	3
Giallo	4
Verde	5
Blu	6
Viola	7
Grigio	8
Bianco	9

Tabella 5 – Codifica del Moltiplicatore (3ª banda nelle resistenze tradizionali, 4ª in quelle di precisione)

Argento	0,01
Oro	0,1
Nero	1
Marrone	10
Rosso	100
Arancio	1K
Giallo	10K
Verde	100K
Blu	1M
Viola	10M

Tabella 6 – Codifica della Tolleranza (4ª banda nelle resistenze tradizionali, 5ª in quelle di precisione)

Argento	±10%
Oro	±5%
Marrone	±1%
Rosso	±2%
Verde	±0,5%
Blu	±0,25%
Viola	±0,1%

• terza fascia: la terza fascia colorata definisce il numero di zeri che è necessario aggiungere al valore ottenuto componendo in sequenza le prime due cifre (**tabella 5**). Essa individua, quindi, un fattore moltiplicativo potenza di dieci. In realtà la terza fascia può anche individuare un fattore di divisione; infatti il colore oro per la terza fascia indica fattore di divisione pari a 10, mentre il colore argento indica fattore di divisione pari a 100;

• quarta fascia: il valore ohmico reale di una resistenza non è mai esattamente pari a quello nominale ma può variare all'interno di una certa fascia di tolleranza intorno al valore nominale. Se la quarta fascia presenta colore oro, la tolleranza è pari al ±5% rispetto al valore nominale, se invece è di colore argento la tolleranza è pari al ±10% (**tabella 6**).

Nello standard a 4 bande (**tabella 8**) si hanno quindi solo due cifre significative che corrispondono alle prime due bande. La terza banda è il fattore di moltiplicazione, mentre l'ultima è la tolleranza. La formula per calcolare il valore ohmico è: (10xA+B)x10^M+/-T, dove A e B sono le cifre corrispondenti alle prime due bande, M l'esponente del moltiplicatore base 10 e T la tolleranza.

Esempi di codifica a quattro bande

Esempio 1: marrone – grigio - rosso - oro
Valore: marrone - grigio →18
Moltiplicatore: rosso → x100
→ 1800 Ω=1,8kΩ
Tolleranza: oro → ±5%
Essendo (5% di 1800) = (1800/100)*5=90, una resistenza di questo tipo ha il valore ohmico reale compreso tra (1800-90) Ω e (1800+90) Ω cioè tra 1710 Ω e 1890 Ω.

Esempio 2: giallo – viola – verde - oro
Valore: Giallo – Viola → 47
Moltiplicatore: Verde → x 100K Ω
→ 4700 K Ω = 4,7 M Ω
Tolleranza: Oro → ±5%

Tabella 7 – Codifica del coefficiente di temperatura (6ª banda), presente solo in resistenze di precisione (ppm/K = variazione di resistenza in ppm (punti per milione) per grado ºK)

Nero	200ppm/K
Marrone	100ppm/K
Rosso	50ppm/K
Arancio	25ppm/K
Giallo	15ppm/K

Esempio 3: Giallo - Viola - Rosso - Oro
Valore: giallo – viola → 47
Moltiplicatore: Rosso → x 100
→ 4700 Ω = 4,7 kΩ
Tolleranza: Oro → ±5%

In **figura 4** sono riportati i valori nominali della sequenza di resistenze standard E12 (tolleranza 10%).

Codifica a cinque o sei bande

Passando dalle comuni resistenze a quelle di "precisione" a strato metallico, la lettura si complica leggermente, perchè sul loro corpo vi sono 5, e in certi casi anche 6, fasce colorate. Tali resistenze sono caratterizzate da tre fasce di colore che indicano le prime tre cifre significative, mentre la quarta fascia indica il numero di zeri. In **tabella 9** è riportato il codice a 5 e 6 bande. In alcuni casi, infatti, viene riportata una ulteriore sesta banda (non molto frequente) per indicare il coefficiente di temperatura. Nello standard a 5 bande si hanno 3 cifre significative che corrispondono alle prime tre bande. La quarta banda è il fattore di moltiplicazione, mentre l'ultima è la tolleranza. La formula per calcolare il valore è (100xA+10xB+C)x10^M +/-T dove A, B e C sono le cifre corrispondenti alle prime tre bande mentre M è il moltiplicatore base 10 e T la tolleranza.

Esempio di codifica a cinque bande

Marrone - Rosso - Marrone - Arancio - Rosso
valore: Marrone-Rosso-marrone → 121
Moltiplicatore: arancio → x 1000
→ 121 kΩ
Tolleranza: Rosso → ±2%

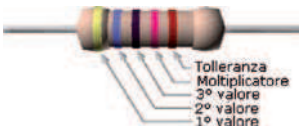
Esempio di codifica a sei bande

Marrone – rosso – viola – nero – oro - rosso
Valore: Marrone - Rosso - Viola → 127
Moltiplicatore: Nero → x 1
→ 127 Ω

Tabella 8 - Codifica dei resistori a 3 o 4 anelli colorati [2]

Colore	1° Anello	2° Anello	3°Anello	4° Anello
	Cifra 1	Cifra2	Moltiplicatore	Tolleranze
-	-	-	-	± 20%
argento	-	-	10^{-2}	± 10%
oro	-	-	10^{-1}	± 5%
nero	0	0	10^{0}	-
marrone	1	1	10^{1}	± 1%
rosso	2	2	10^{2}	± 2%
arancio	3	3	10^{3}	-
giallo	4	4	10^{4}	-
verde	5	5	10^{5}	± 0,5%
blu	6	6	10^{6}	± 0,25%
viola	7	7	10^{7}	± 0,1%
grigio	8	8	10^{8}	± 0,05%
bianco	9	9	10^{9}	-
	Esempio: (Marrone=1), (Nero=0), (Arancio=3), 10 x 1000 = 10kohm, Tolleranza (Oro) = ±5%			

Tabella 9 - Codifica dei resistori a 5 o 6 anelli colorati [2]

Colore	1° Anello	2° Anello	3°Anello	4° Anello	5° Anello	6° Anello
	Cifra 1	Cifra2	Cifra3	Moltiplicatore	Tolleranza	Coeff. di temperatura
-	-	-	-	-	± 20%	-
argento	-	-	-	10^{-2}	± 10%	-
oro	-	-	-	10^{-1}	± 5%	-
nero	0	0	0	10^{0}	-	200 ppm/K
marrone	1	1	1	10^{1}	± 1%	100 ppm/K
rosso	2	2	2	10^{2}	± 2%	50 ppm/K
arancio	3	3	3	10^{3}	-	15 ppm/K
giallo	4	4	4	10^{4}	-	25 ppm/K
verde	5	5	5	10^{5}	± 0,5%	-
blu	6	6	6	10^{6}	± 0,25%	10 ppm/K
viola	7	7	7	10^{7}	± 0,1%	5 ppm/K
grigio	8	8	8	10^{8}	± 0,05%	-
bianco	9	9	9	10^{9}	-	-
	Esempio: (Giallo=4), (Violetto=7), (Nero=0), (Rosso=2), 470 x 100 = 47kohm, Tolleranza (Marrone) = ±1%					

Tolleranza: Oro → ±5%
Coefficiente di temperatura: Rosso à 50ppm/K

Alcune osservazioni pratiche sull'impiego del codice dei colori

Una delle difficoltà che trova chi si avvicina per la prima volta alla realizzazione di un circuito elettronico è quella di leggere correttamente il valore delle resistenze che sta maneggiando. Il più delle volte si è portati a misurare il valore ohmico della resistenza mediante l'impiego di un multimetro. Un vero tecnico, tuttavia, non ha bisogno di un multimetro poiché conosce il codice dei colori ed è capace di applicar-

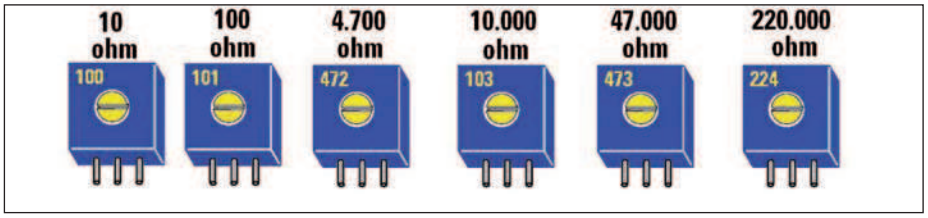


Figura 7: nei Trimmer il valore ohmico è spesso espresso mediante tre cifre. Le prime due sono quelle significative mentre la terza indica quanti zeri aggiungere per ottenere il valore nominale (100 significa quindi 10 Ω, 101 significa 100 Ω, 472 significa 4.700 Ω e così via).

Figura 8: potenziometro da pannello con manopola (a sinistra) e da circuito stampato con regolazione a slitta (slider-al centro) o rotativa (a destra) [2].

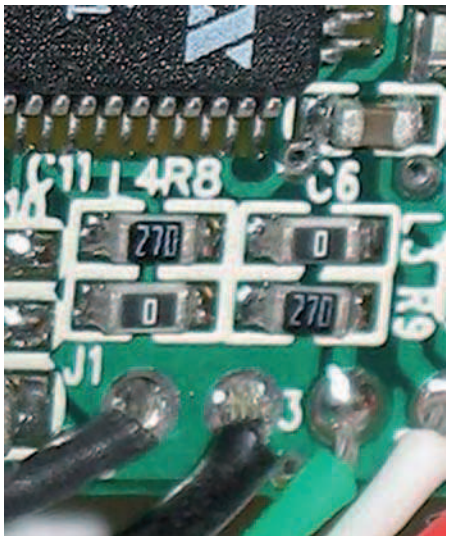


Figura 9: esempio di scheda con componenti SMD a bordo [2].

Tabella 10 – Dimensione standard dei package dei componenti SMD [2]

PACKAGE=	DIMENSIONI IN POLLICI (LUNGHEZZA X LARGHEZZA)	= DIMENSIONI IN MM
"01005"	= 0,016" × 0,008"	= 0,4 mm × 0,2 mm
"0201"	= 0,024" × 0,012"	= 0,6 mm × 0,3 mm
"0402"	= 0,04" × 0,02"	= 1,0 mm × 0,5 mm
"0603"	= 0,063" × 0,031"	= 1,6 mm × 0,8 mm
"0805"	= 0,08" × 0,05"	= 2,0 mm × 1,25 mm
"1206"	= 0,126" × 0,063"	= 3,2 mm × 1,6 mm
"1210"	= 0,12" x 0,10"	= 3,2 mm x 2,5 mm
"1812"	= 0,18" × 0,12"	= 4,6 mm × 3,0 mm
"2010"	= 0,20" x 0,10"	= 5,0 mm x 2,5 mm
"2512"	= 0,25" × 0,12"	= 6,3 mm × 3,0 mm

Tabella 11 – Esempi di codifica del valore ohmico di resistori SMD

"334"	= 33 × 10.000 ohm = 330 kohm
"222"	= 22 × 100 ohm = 2,2 kohm
"473"	= 47 × 1.000 ohm = 47 kohm
"105"	= 10 × 100.000 ohm = 1 Mohm

lo in maniera corretta e celere. Se siete ai primi passi con i circuiti elettronici, pertanto, tenete sempre a disposizione il codice dei colori e ogni volta che vi trovate nella condizione di dover conoscere il valore della resistenza che avete tra le mani, applicatelo; con il tempo vi diventerà famigliare e di semplice applicazione.

Come accennato prima, le resistenze elettroniche il cui valore è codificato secondo il codice dei colori, possono avere 4, 5 oppure 6 bande colorate. Fisicamente, una resistenza avendo una forma geometricamente simmetrica può far sorgere dubbi su quale sia l'estremo del componente da cui cominciare a leggere il codice. Per individuare il primo anello, si deve partire da quello più vicino a uno dei terminali metallici: non sempre ciò è agevole. In caso di dubbio, si possono fare alcune prove, prima partendo da un lato, poi dall'altro, nel conteggiare il primo anello: si possono trovare valori ragionevoli oppure no e dedurre di conseguenza se si sta orientando nel giusto verso il componente o nel verso opposto.

Un altro dettaglio utile allo scopo si deduce facendo riferimento alle precedenti tabelle: si può osservare come la fascia colore oro e quella di colore argento non portano mai significato di cifra ma di moltiplicatore o tolleranza. Di conseguenza, se nella sequenza delle fasce ve ne è una di colore oro o argento (come accade nella maggior parte dei casi) è sufficiente cominciare a leggere il codice dal lato opposto. In ogni caso occorre posizionare la resistenza con i valori di tolleranza (generalmente i colori oro o argento) sulla destra.

Un'altra cosa che occorre osservare è che le bande, generalmente, non sono centrate rispetto al corpo del componente, pertanto si orienta questo in modo da avere a sinistra il terminale più vicino alle bande. Se tuttavia le bande appaiono centrate, una che si trova agli estremi generalmente è più larga delle altre: il componente va orientato in modo da vedere questa banda a destra.

Esistono poi situazioni in cui l'identificazione delle fasce colorate può apparire incerta anche per altri motivi: per esempio la quarta fascia poco evidente o cancellata oppure non si riesce a distinguere in maniera netta un colore da un altro, per esempio il rosso dall'arancio. In questi casi il problema può essere agevolmente superato ricordando ancora una volta che il valore ohmico che si deve ottenere dalla decodifica della sequenza deve in ogni caso risultare un valore standard.

Un altro dettaglio che può aiutare l'interpretazione del codice è quello che vede l'ultima banda (che può essere la 4ª o la 5ª) più distanziata dalle altre.

Per chi vuole memorizzare l'ordine del codice dei colori facilmente suggerisco di

MARCHEFIERE

ERF ENTE REGIONALE PER LE MANIFESTAZIONI FIERISTICHE

Quartiere Fieristico CIVITANOVA MARCHE (MC)

6-7 MARZO 2010

Orario:
9-13
15-19,30

30ª Mostra Mercato Nazionale

Radiantistica Elettronica

Materiale radiantistico per C.B. - Antenne e Parabole per radioamatori
Apparecchiature per telecomunicazioni - Surplus - Elettronica
Telefonia - Computers - e TV sat - Hi-Fi
Radio d'epoca - Editoria specializzata

9º Mercatino del Radioamatore

Libero scambio tra privati di apparecchiature amatoriali

in collaborazione con Associazione Radioamatori Italiani
Sezione di Civitanova Marche

VOLONTARI EMERGENZA RADIO ECO RADIO 60

DISCO

Mostra mercato del disco usato in vinile
e CD da collezione

info

ERF • ENTE REGIONALE PER LE MANIFESTAZIONI FIERISTICHE
Quartiere Fieristico di Civitanova Marche • Tel. 0733 780815 • Fax 0733 780820
www.erf.it - segreteria.radiantistica@erf.it

CODICE MIP 2754050

Tabella 12 – Esempi di codifica di resistori SMD di valore inferiore a 100 ohm

"100"	= 10 × 1 ohm = 10 ohm
"220"	= 22 × 1 ohm = 22 ohm
"470"	= 47 × 1 ohm = 47 ohm

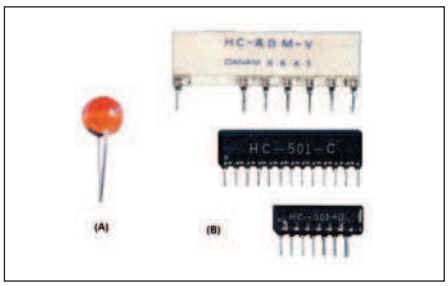


Figura 10: esempi di NTC (A) e di reti resistive (B) [1].

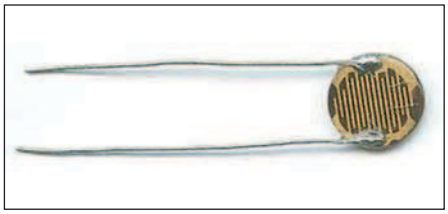


Figura 11: fotoresistore [2].

Figura 12: resistenza 0 Ω [2].

ricordare questa breve frase mnemonica trovata sul web: Non Metterti Rubicondo Alla Guida: Vino e Birra Van Giù Bene. Le iniziali delle parole sono anche le iniziali dei colori esattamente nella giusta sequenza.

Altri tipi di codifiche del valore ohmico

Il codice dei colori è generalmente applicato per resistenze elettronici di bassa potenza (tipicamente 1/4W, 1/8W, 1W o 2W) che sono anche quelle fisicamente più piccole. Le resistenze di potenza maggiore e di piccolo valore ohmico, generalmente resistenze a filo, non impiegano il codice dei colori ma presentano spesso il valore ohmico direttamente stampato sul corpo (**figura 5**). Le situazioni più comuni che possono presentarsi sono:

Tabella 13 – Esempi di codifica del valore di resistori SMD di precisione

"1001"	= 100 × 10 ohm = 1 kiloohm
"4992"	= 499 × 100 ohm = 49,9 kiloohm
"1000"	= 100 × 1 ohm = 100 ohm

- CASO 1: sul corpo appare scritto qualcosa del tipo 0,15 ohm oppure 1,5 ohm o ancora 15 ohm. In questo caso vi sono pochi dubbi, quello scritto è inequivocabilmente l'effettivo valore ohmico nominale del componente;
- CASO 2: sul corpo appare un valore numerico preceduto dalla lettera R. In questo caso tale lettera equivale ad anteporre alle cifre che la seguono "0," Pertanto R01 corrisponde a 0,01 Ω , R15 corrisponde a 0,15 Ω, R1 equivale a 0,1 Ω e così via;
- CASO 3: sul corpo appare la lettera R compresa tra cifre numeriche. In questo caso la lettera R equivale alla virgola. Per esempio 4R7 equivale a scrivere 4,7 Ω, 1R2 equivale a 1,2 Ω e così via.

Resistori variabili: i trimmer

Spesso nei circuiti elettronici occorre disporre di resistori il cui valore ohmico possa essere variato gradualmente da 0 Ω fino a un determinato valore massimo: questi componenti prendono il nome di trimmer e sono raffigurati con lo stesso simbolo di una resistenza cui è aggiunto il simbolo di cursore (**figura 6**). Per essere più precisi, resistori variabili possono essere divisi in due categorie: quelli regolabili manualmente (trimmer e potenziometri) e quelli che variano il proprio valore ohmico in funzione di un altro parametro fisico, tipicamente la temperatura (termo-resistori) o l'intensità luminosa della luce che li colpisce (foto-resistori). I trimmer e i potenziometri sono di frequente impiego in molte apparecchiature elettroniche di uso comune come radio, televisori, amplificatori, mixer, equalizzatori e in tutti i circuiti in cui è necessario operare sintonizzazioni, tarature o regolazioni. L'elemento fondamentale di questi componenti è un conduttore mobile strisciante sull'elemento resistivo attraverso il quale è possibile variare il valore della resistenza offerta dal componente stesso ai suoi morsetti. La regolazione può avvenire a seconda dei casi in due modi: agendo su un cursore rotativo oppure su di un cursore lineare. Da un punto di vista costruttivo possono essere realizzati a strato, a filo, doppi, con o senza interruttore e possono presentare variazione resistiva lineare o logaritmica.

Mentre i potenziometri e i trimmer impiegati per regolazioni molto frequenti (per esempio la regolazione del volume di un apparato audio) sono tipicamente montati a pannello e sono dotati di opportuna manopola, quelli che servono per regolazioni saltuarie, per esempio tarature, non sono dotati di manopola, richiedono l'uso di un giravite per agire su una apposita vite di regolazione e sono, generalmente, montati direttamente sul circuito stampato dell'apparecchiatura. Un'altra distinzione che è possibile fare è quella tra trimmer a singolo giro e trimmer di precisione multigiro in cui la vite di regolazione consente di regolare il valore resistivo voluto con molta precisione su un range composto da più giri.

Spesso nei trimmer è impiegata una codifica del valore ohmico differente da quelle descritte fino a ora. Generalmente sul corpo del trimmer compare una sequenza di cifre che non rappresenta il reale valore ohmico massimo del trimmer. Il codice che esprime il valore ohmico del componente è di tipo numerico ed è composto da tre cifre: l'ultima cifra indica quanti zeri bisogna aggiungere al valore composto con le cifre precedenti per ottenere il valore ohmico nominale. Se per esempio sul corpo del trimmer è scritto 101, il valore ohmico massimo del trimmer corrisponde a 10, cui far seguire uno zero, quindi 100 Ω e non 101 Ω, come erroneamente si potrebbe essere portati a pensare. Analogamente, se sul corpo del trimmer c'è scritto 224, il valore ohmico del trimmer è pari a 22, cui far seguire 4 zeri, e quindi 220000 Ω e non 224 Ω. In **figura 8** sono riportati alcuni esempi di questo tipo di codifica che dovrebbero essere sufficienti a dissipare qualunque dubbio in merito.

Tipi di trimmer

Vi sono in commercio trimmer di diversa forma e dimensione, anche da un punto di vista geometrico, tali da poter essere montati sia orizzontalmente che verticalmente sul circuito stampato (**figura 8**). General-

mente, il morsetto centrale rappresenta quello elettricamente connesso al cursore. Collegando uno dei due terminali estremi con il morsetto di cursore, si cortocircuita parte dell'intera resistenza ohmica; questa parte può essere variata agendo sul cursore e in tal modo è possibile ottenere il valore ohmico voluto. Questi componenti possono essere, in ogni caso, impiegati anche senza cortocircuitare il morsetto di cursore con uno di quelli estremi: in questo caso il componente è impiegato come potenziometro. Se non riuscite a intuire immediatamente l'importanza di questi componenti, pensate a quante volte vi è capitato di agire sulla manopola di una radio, variandone in questo modo il volume oppure sulle manopole di un equalizzatore variando il peso degli alti oppure dei bassi rispetto alle altre frequenze.
Oggi queste applicazioni sono meno comuni dal momento che la maggior parte dei controlli e delle regolazioni che si effettuano sulle apparecchiature sono di tipo digitale, tuttavia tali componenti restano di diffuso impiego in elettronica. Tra le diverse caratteristiche di un potenziometro vi è la legge di variazione del valore ohmico rispetto alla posizione angolare della manopola nel caso di potenziometro rotativo, o della posizione lineare del cursore nel caso di potenziometro a slitta. Tale legge di variazione può essere di natura lineare oppure logaritmica. Se vi state chiedendo a cosa possa servire un potenziometro logaritmico pensate al fatto che il nostro udito presenta una sensibilità nei confronti della potenza sonora di tipo logaritmico; ciò significa che per percepire un raddoppio della potenza sonora è necessario che questa venga realmente quadruplicata. Ciò spiega il motivo per cui tutti i potenziometri impiegati nelle apparecchiature audio per la regolazione del volume sono generalmente di tipo logaritmico.

Resistori SMD

I resistori elettronici realizzati in tecnologia discreta mostrati fino a ora non sono gli unici impiegati in elettronica; vi sono resistori che si prestano al montaggio automatico e che permettono una maggiore integrazione dei componenti sulla board. Si tratta dei resistori realizzati in tecnologia SMD. L'immagine in **figura 9** mostra quattro resistori SMD (il componente nell'angolo in alto a destra, con il corpo di colore marrone, è un condensatore). Due resistori hanno valore di 0 ohm, gli altri due possono essere da 27 ohm o 270 ohm (la codifica dei costruttori può variare). I resistori SMD (acronimo di Surface Mounting Device) o SMT (Surface-Mount Technology) hanno l'aspetto di minuscole piastrine rettangolari di dimensioni variabili a seconda della potenza. Le dimensioni più comuni sono (lo standard è usato anche per i condensatori) quelle riportate in **tabella 10**. Il montaggio è automatizzato: sul circuito stampato viene posto il materiale saldante sulle due piazzole di saldatura, mediante una macchina automatica programmata vengono quindi posti i componenti SMD nelle esatte posizioni della pasta saldante; segue il passaggio in un apposito forno che sottoponendo la piastra e i componenti posizionai sulla stessa ad un opportuno riscaldamento, secondo un preciso profilo temporale di temperatura, ne determina la saldatura.

Lettura dei resistori a montaggio superficiale (SMD)

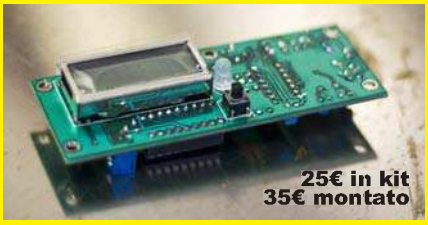
I resistori a montaggio superficiale sono generalmente marcati con un codice a tre cifre: le prime due sono cifre significative, la terza indica il numero degli zeri da aggiungere a queste per ottenere il valore ohmico nominale (**tabella 11**). La stessa codifica viene utilizzata anche per i condensatori a montaggio superficiale, per i quali il valore letto è da intendersi in pF.
Resistori di valore inferiore a 100 ohm sono marcati con una sequenza di tre cifre di cui l'ultima è uno zero cioè dieci elevato alla potenza zero cioè uno (**tabella 12**).I resistori di precisione sono invece siglati con un codice a quattro cifre. Le prime tre sono cifre significative mentre la quarta indica la potenza di dieci come mostrato negli esempi in **tabella 13**.

Reti resistive

Oltre ai resistori individuali, esistono componenti che integrano all'interno delle ve-

micromed
www.micromed.it
COMPONENTI CNC
STEPPER E DRIVERS
MICROCONTROLLORI
ALIMENTATORI PC TAPE
PC FONO TELECAMERE
RDS DECODER DISPLAY
LCD MOTORI CC CIRCUITI
INTEGRATI AUDIO VIDEO
OCCASIONI MINI ROBOT
vendita per corrispondenza

M22
Motore Stepper
4 Fasi - 2A - 1,1 Ohm - 200 Passi
Prova il nuovo portale eshop!
eshop.micromed.it

RDS Decoder

25€ in kit
35€ montato
Dispositivo per l'acquisizione e l'elaborazione dei messaggi RDS delle stazioni radio FM. Applicabile a qualsiasi ricevitore anche portatile per individuare il nome della stazione e leggere trasmissioni di testo.
Completo di display 1x 8 e con la possibilità di aggiungere un display esterno 2x16 o 2x20.
CODICE MIP 2777766

re e proprie "reti resistive", costruite con la tecnica chiamata "a film spesso", mediante un processo di metallizzazione su una sottile piastrina di ceramica (**figura 10**).

Resistori Custom

Esistono anche applicazioni in cui il resistore deve avere caratteristiche particolari; tipico è il caso degli strumenti di misura elettronici. Se in commercio non è disponibile il resistore con dimensioni, valore e precisione necessaria per l'impiego previsto, il costruttore dello strumento progetta da sé il resistore abbinandogli una sigla; questo risulta quindi un componente custom non in commercio che può, di conseguenza, essere fornito come ricambio solo dal costruttore dello strumento.

Le fotoresistenze

In elettronica esistono anche diversi componenti che non sono delle classiche resistenze ma piuttosto componenti in grado di variare il proprio valore ohmico in funzione dell'intensità di una determinata grandezza fisica rispetto alla quale risultano sensibili. Uno di questi è la fotoresistenza, cioè letteralmente resistenza sensibile alla luce. È facile verificare come il valore ohmico offerto ai propri terminali da una fotoresistenza possa variare da valori dell'ordine del megaohm (1-10MΩ) in condizioni di buio, a valori di poche decine di ohm in condizioni di forte intensità luminosa incidente. Per avere un'idea immediata di alcune applicazioni di componenti di questo tipo, si pensi agli automatismi di apertura e chiusura di un ascensore o di porte automatiche, oppure a sistemi di conteggio automatico di pezzi in una catena di montaggio o di produzione o ancora alla misura stessa dell'intensità luminosa.

Termoresistori (NTC e PTC)

I termoresistori o termistori sono invece resistori che presentano elevati valori del coefficiente di temperatura; essi si distinguono in PTC ed NTC (**figura 10**). I resistori PTC (Positive Temperature Coefficient) hanno un coefficiente di temperatura positivo, ossia aumentano la loro resistenza con l'aumento della temperatura mentre gli NTC (Negative Temperature Coefficient) presentano un coefficiente di temperatura negativo (tipicamente tra -6% e -2% per grado centigrado), ossia riducono la loro resistenza con l'aumentare della temperatura. Essi sono impiegati o per la misura della temperatura o come sensori nei sistemi di controllo.

PER *approfondire...*

[1] Scuola di elettronica, Teoria, tecnica, pratica (Enciclopedia Peruzzo Editore)

[2] Resistore, http://it.wikipedia.org

[3] G. Biondo, E. Sacchi, Componenti elettrici passivi (Normalizzazione e codificazione), Manuale di Elettronica e Telecomunicazioni, cap IX, 4° ed., Hoepli

[4] Tabella dei colori dei resistori, www.elettronicando.net

[5] www.conversioni.it

[6] Electronic color code, http://it.wikipedia.org

[7] www.themeter.net

I resistori 0 Ω

Pochi sanno che in commercio esistono delle resistenze che presentano lo stesso aspetto delle comuni resistenze elettroniche e valore ohmico pari a zero Ω. Il valore è indicato con un unico anello nero (**figura 12**). Nel caso di realizzazione SMD questo può essere siglato "000" o "0000". L'uso di resistori da zero ohm è utile in molti casi:

- in alternativa a un componente previsto ma non usato in alcune applicazioni della scheda elettronica, si può prevedere, in parallelo, un resistore da zero ohm senza dover rifare il circuito stampato;
- nell'uso di circuiti integrati che possono avere vari modi di funzionamento, si mantiene aperta la possibilità di montare un resistore opportuno (compreso zero ohm) senza dover rifare il circuito stampato;
- in schede elettroniche particolarmente complesse (specialmente di tipo monofaccia), un resistore 0Ω permette il passaggio di piste sotto di sé, facilitando lo sbroglio del circuito durante la progettazione ed evitando l'impiego di ponticelli filari durante la realizzazione.

Un problema che pongono i resistori da zero ohm è la dissipazione di potenza non calcolabile con la legge di Ohm (nei dati tecnici, generalmente viene indicata la massima corrente che il resistore può sopportare senza danneggiarsi).

Misura di resistenze saldate su una scheda elettronica

Nella pratica ci si può trovare a dover misurare valori di resistori già montati in un circuito senza poterli dissaldare e senza poter sezionare le piste che ne realizzano la connessione agli altri componenti. Usando un multimetro (con il circuito non alimentato), si legge un valore che pone almeno un limite minimo al valore effettivo della resistenza in questione, in quanto in parallelo al resistore vi sono altri elementi resistivi che abbassano la resistenza del ramo. Rifacendo la misura scambiando i terminali (la presenza di componenti attivi e diodi ha un comportamento nella misura diverso a seconda del verso della corrente che il multimetro imprime nella misura), si ottiene un secondo valore. Dopo questa seconda misura si sceglierà il valore più alto tra quelli misurati, valore che in ogni caso è solo un limite minimo all'effettivo valore del componente.

Alcuni utili strumenti

Chiudiamo questo articolo, dedicato ai componenti resistivi comunemente impiegati in elettronica, segnalando, soprattutto a neofiti e principianti che non vogliano perder tempo a decifrare il valore di una resistenza decodificandone il codice dei colori, o che desiderino un ausilio didattico per prendere dimestichezza con questo tipo di operazioni, che in rete sono disponibili diversi applicativi freeware utili allo scopo. ❑

CODICE MIP 2757063

POSCOPE BASIC

Uno strumento indispensabile

6 STRUMENTI IN 1!

1. Oscilloscopio 2 canali
2. Analizzatore di spettro 2 canali
3. Registratore 2 canali
4. Analizzatore logico 16 canali
5. Generatore logico 8 canali
6. Generatore di segnali PWM a 5 canali

OSCILLOSCOPIO ED ANALIZZATORE DI SPETTRO

Numero canali: 2
Frequenza di campionamento: 100 Hz ÷ 200 KHz
Memoria:
- Buffer di lettura: 1126 campioni/canale (1 canale), 563 campioni/canale (2 canali).
- Pipe di lettura: 64K campioni/canale (1 o 2 canali).

Massima tensione di ingresso: -20 ÷ +20 V
Risoluzione ADC: 10 bits
Triggering:
- Assoluto (per fronti di salita/discesa)
- Differenziale (per differenza tra campioni consecutivi)
- Esterno (per fronti di salita/discesa di segnali TTL)

Funzionalità disponibili: Hamming, Hanning, Blackman, Blackman-Harris.

ANALIZZATORE LOGICO

Numero canali: 16 (8 se utilizzato il generatore logico)
Frequenza di campionamento: 1 KHz ÷ 8 MHz
Memoria:
- Buffer in lettura (Fs=4-8 MHz) 128 bit/canale.
- Buffer in lettura (Fs=2-2.66 MHz) 1160 bit/canale.
- Buffer in lettura (Fs<=1 MHz) 1544 bit/canale
- Buffer in lettura (in mod. concatenamento) 1 Mbit/canale.
- Pipe di lettura (Fs < 500KHz) 4K a 256 Mbit/canale.

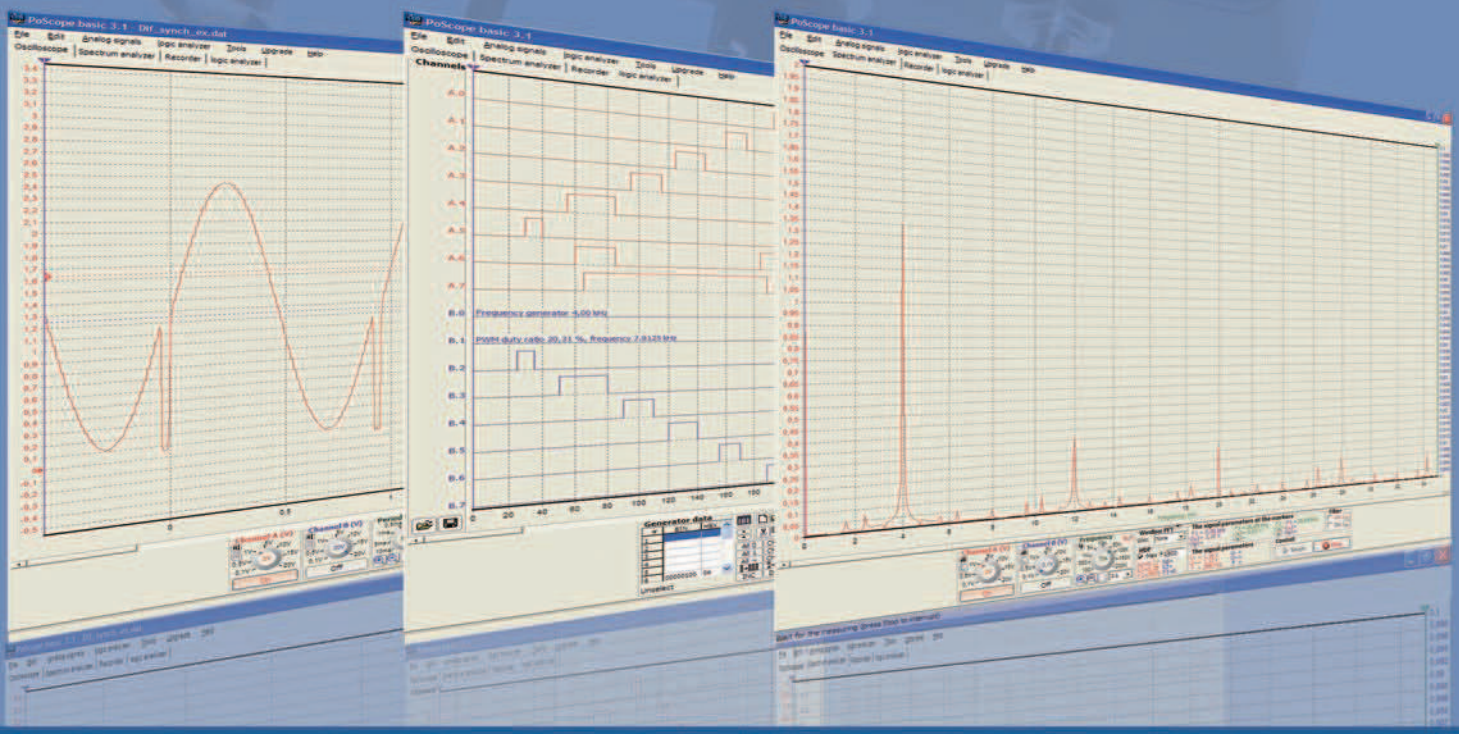
Massima tensione di ingresso: 0 ÷ +5 V
Triggering: per fronti del segnale, maschere, impulsi persi, clock esterno.
Clock: interno/esterno

REGISTRATORE

Frequenza di campionamento: 0.01 Hz ÷ 200 KHz
Capacità massima di registrazione: 24 ore (Fs < 100 Hz)
Tensione d'ingresso: -20 ÷ +20 V (hardware 2 sub-band)
Risoluzione ADC: 10 bit

GENERATORE LOGICO

Numero canali: 8
Frequenza di campionamento: 1 KHz ÷ 1 MHz
Memoria: 1544 bit/canale
Tensione di uscita: "0" - 0 V, "1" - 3.3 V
Massima corrente in ingress/uscita: 10 mA



CODICE MIP 2779082

Ordinalo subito su www.ieshop.it/poscope

l'angolo di Mr A.KEER

DECODER "con memoria"

Con i decoder "con memoria" si conclude la rassegna dei principali dispositivi in grado di controllare i visualizzatori formati da digit "a 7 segmenti". Scopriamone i segreti e il modo migliore per utilizzarli

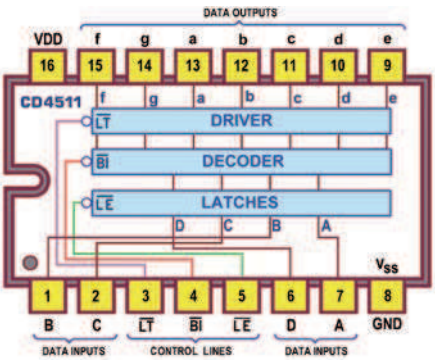


Figura 1: pinout BCD to 7-segment Latch/Decoder/Driver 4511.

In chiusura della puntata precedente abbiamo constatato la concreta realtà di disporre di super decoder "a 7 segmenti" (dispositivi combinatori) dotati di uno strato sequenziale, cioè della possibilità di fruire su ciascuno dei propri ingressi di un flip-flop (un elemento di memoria) e di uno strato di potenza (driver) tale da garantire alle rispettive uscite le migliori condizioni elettriche per pilotare un LED. A questa categoria appartengono i sofisticati 4511 e 9368, oggetto della nostra analisi attuale.

DECODER DA BCD A 7 SEGMENTI: 4511, 4513

Come tutti i decoder per digit analizzati finora, anche il 4511 (unico, con l'analogo 4513, disponibile nella serie CMOS per il controllo di digit a LED) appartiene alla categoria dei BCD to 7-segment Decoder, caratterizzata dalla capacità di interpretare solo 10 delle possibili 16 parole del codice binario a 4 bit proposto sulle sue 4 linee d'ingresso. Come il TTL 74LS48 (col quale è in parte pin-out compatibile, escludendo 3 piedini di controllo), il decoder 4511 converte il codice BCD proposto sulle sue linee d'ingresso, generando sulle linee d'uscita dei segnali attivi alti: i segmenti del digit a esso collegato si accenderanno dunque con un 1 logico, per cui questo componente è adatto a digit a catodo comune; il suo pin-out è illustrato in **figura 1**. Osservando la figura appare subito evidente che si tratta di un componente speciale, in grado di dare molto di più di quanto ci si aspetta, cioè di svolgere tre funzioni in una: consente di congelare in una memoria a 4 bit le informazioni fornite sulle linee d'ingresso; garantisce ovviamente il suo intrinseco servizio di decoder; è dotato di driver su ciascuna delle 7 linee d'uscita, al fine di fornire le migliori condizioni di pilotaggio ai LED a esse collegati.
Va sottolineata l'impossibilità di gestire più componenti 4511 in cascata con la tecnica *Ripple Blanking* garantita invece dal 74LS48 (per altro chiamato a svolgere lo stesso servizio combinatorio), descritta la volta scorsa e ripresa più avanti per il 9368: la disponibilità del segnale di controllo RBO (*Ripple Blanking Output*) è stata sacrificata per far posto a quello LE (*Latch Enable*) necessario per governare una memoria.

Lo strato di memoria

Prima di raggiungere il decoder, ciascuna delle 4 linee d'ingresso attraversa un flip-flop di tipo *D-Latch*; l'utilizzo di questo tipo di memoria non è molto diffuso nel progetto digitale e questo è uno dei casi più peculiari. I suoi 4 elementi sono attivati sul livello della linea di sincronismo (LE, *Latch Enable*, *pin5*), che li controlla contemporaneamente: quando LE è a livello 0, fissato al negativo dell'alimentazione, i 4 bit del codice BCD applicato in ingresso passano inalterati a valle della memoria, che risulta essere trasparente (come se non ci fosse) nei loro confronti, così come fosse una porta aperta (l'uscita insegue l'ingresso); nell'istante in cui LE passa da un livello basso a uno alto (cioè sul fronte di salita di LE) le uscite *D-Latch* "scattano" sul codice binario presente su di esse in quel momento, tenendolo bloccato (memorizzandolo) per tutto il tempo nel quale la linea LE è tenuta a livello 1, fino a quando verrà riportata a 0. La notazione logica mette in evidenza queste specifiche (graficamente, con la presenza del pallino sulla linea d'ingresso e formalmente soprasegnando il segnale LE): in condizioni di funzionamento normale (oppure se l'azione della memoria non è desiderata) questo *pin5* va posto a massa! Si tratta di uno dei più frequenti errori dei giovani virgulti che, dimenticandolo scollegato, si disperano perché il digit collegato al decoder rimane "bloccato" anche quando è attivo un eventuale conteggio in ingresso.

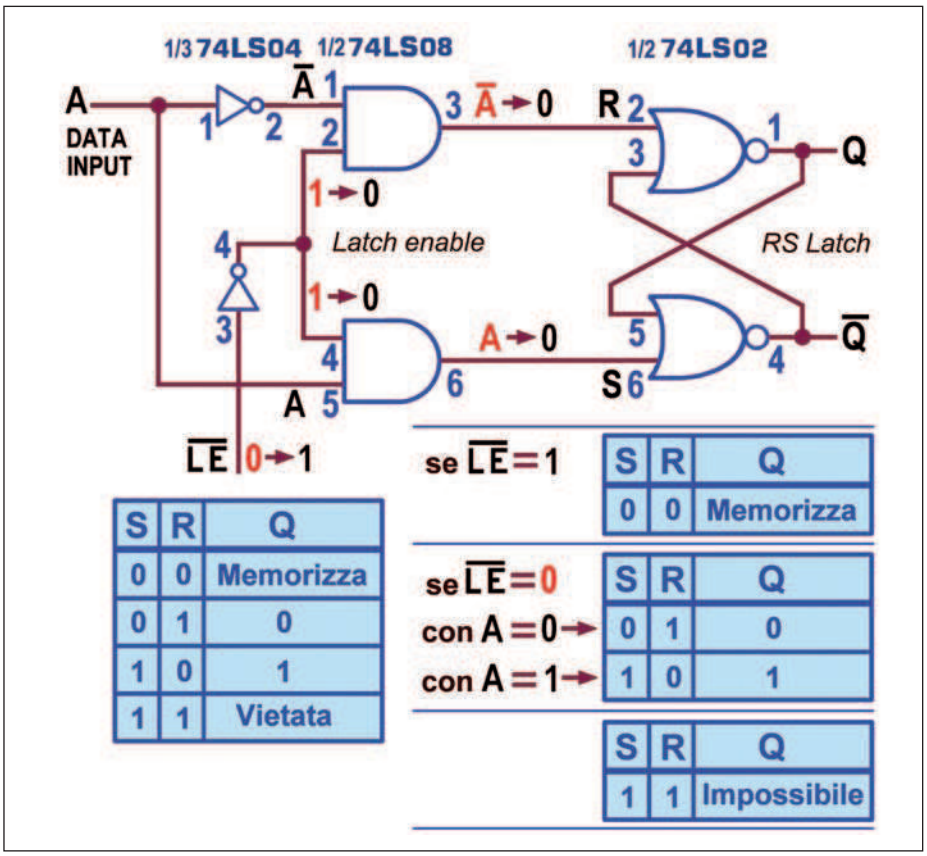


Figura 2: schema di un D-Latch BCD to 7-segment Latch/Decoder/Driver 4511.

Va detto che la notazione attiva bassa per LE può generare qualche insicurezza se la si intende come comando per "eseguire la memorizzazione" degli ingressi: di fatto essa avviene quando LE è a 1 così da poter supporre che l'atto si compia per un'azione attiva alta! In realtà il termine *Latch Enable* va inteso come comando per "autorizzare il passaggio del dato sul *latch*", lasciando intendere che quest'ultimo sia abilitato a compiere la sua funzione (quella di memorizzare il dato stesso) solo se il segnale LE tiene aperte le porte d'accesso al *latch*, cosa possibile appunto se esso è forzato a 0, per la presenza di un inverter interno come si vede dallo schema (**figura 2**) valido per ciascuno dei 4 flip flop *D-Latch* utilizzati dal componente.

Resta da valutare l'utilità di questa memoria. Se questo componente è collegato a valle di un contatore (per esempio una Decade CMOS 4029 *Binary/Decade Up/Down Counter*), potrà disporre in ingresso di un codice a 4 bit riproposto ciclicamente dal valore minimo, $(0000)_2$, a quello massimo, $(1001)_2$, con cadenza imposta dalla frequenza dell'onda quadra applicata (per esempio ogni secondo, se il clock è di 1 Hz). Mantenendo a massa l'ingresso LE del decoder, il digit a esso collegato mostrerà le corrispondenti cifre decimali, variabili con la stessa cadenza; non appena LE viene scollegato da massa (per cui sente un 1) il numero sul digit non subirà più aggiornamenti, sebbene il conteggio in ingresso continui regolarmente.

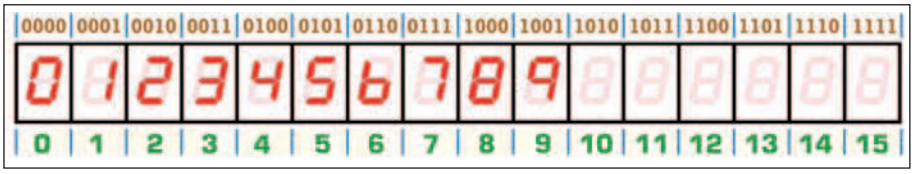


Figura 3: simboli visualizzati con FND500, BCD to 7-segment Latch/Decoder/Driver 4511.

Intervenendo in questo modo sulla linea LE (per esempio con un controllo a fotocellula posto in un dato punto di un percorso sportivo, nel progetto di un cronometro digitale), è possibile generare l'effetto intertempo, congelando il tempo corrente per qualche istante. Mantenendo invece stabilmente alto l'ingresso LE, ogni volta che si fornisce su di esso un breve impulso basso il numero sul digit cambierà istantaneamente in funzione del codice presente in quel momento sugli ingressi DCBA, rimanendo invariato fino al successivo impulso. Questo artificio torna molto utile in alcune applicazioni, per esempio per consentire la visualizzazione del valore misurato da uno strumento digitale (come un frequenzimetro), attivandola solo alla fine di ogni ciclo del conteggio. In questo modo si evita di renderla illeggibile, come sarebbe se il valore fosse mostrato mentre il conteggio è in corso.
In sostanza la misura della frequenza si esegue contando i fronti attivi del segnale da misurare in un intervallo di grande precisione fissato dalla base dei tempi e mostrandone il valore solo alla fine.

La funzione intrinseca di decoder

Come i suoi simili TTL anche questo componente accetta solo le prime 10 combinazioni, da $(0000)_2$ a $(1001)_2$, appartenenti al codice BCD. Se, nonostante il divieto, si fornisce una delle ultime 6 combinazioni vietate, i componenti TTL visti la scorsa puntata davano corpo a simboli sostanzialmente casuali, dovuti alle scelte imposte dal progetto combinatorio del componente, con questo invece si otterrà come effetto quello di spegnere il digit controllato dal decoder. Da notare che le stesse scelte di progetto impongono anche al 4511 la visualizzazione del 6 $(0110)_2$ e del 9 $(1001)_2$ rispettivamente priva del trattino in alto o in basso, tipica del TTL 74LS48, preso sempre come oggetto di confronto. La **figura 3** riassume l'aspetto del digit collegato al decoder, in ogni situazione. Su questa parte circuitale agisce l'influenza del segnale d'ingresso BI (*Blanking Input, pin4*), attivo basso; se su di esso si forza uno 0, tutte le linee d'uscita di segmento sono forzate a 0, causando lo spegnimento di tutti i segmenti del digit. Questa linea di controllo ha un livello di priorità secondo solo al segnale

d'ingresso LT per cui potrà produrre l'effetto descritto solo se LT non è contemporaneamente attivo. Curiosamente nel 74LS48 succede esattamente l'opposto: BI è prioritario rispetto a LT (e a ogni altro ingresso). Anche con questo componente è possibile controllare l'intensità luminosa del suo digit con la tecnica di *blanking* (spegnimento) descritta per i suoi simili TTL nella puntata precedente: basta applicare sul pin4 un'onda quadra non simmetrica e modificare il suo duty cycle (rapporto tra la durata della parte alta e il periodo del segnale) purché la frequenza sia maggiore di 45 Hz (per i digit con LED rossi, all'arseniuro fosfuro di gallio, GaAsP) al fine di garantire una sequenza di spegnimenti (imposti da BI attivo, cioè a 0) e accensioni (imposti da BI disattivo, cioè a 1) sufficientemente veloce da essere percepita come variazione di luminosità per effetto della persistenza ottica dell'immagine sulla retina dall'occhio umano.
La **figura 4** (già proposta la volta scorsa) ricorda l'aspetto dei segnali necessari rispettivamente per una discreta (duty cycle pari a 86%) e una bassa (duty cycle del 14%) accensione dei digit; l'effetto ottenuto è legato al valore medio del segnale imposto su BI, più vicino a 1 logico nel primo caso e allo 0 nel secondo.

Lo strato del driver

Il terzo strato di hardware è proprio una gran cosa: ciascuna delle uscite interne del decoder è dotata di uno stadio realizzato

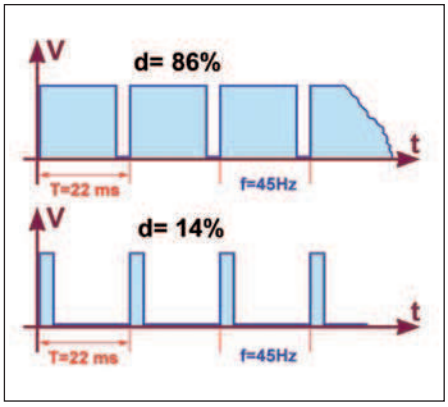


Figura 4: effetto Blanking, BCD to 7-segment Latch/Decoder/Driver 4511.

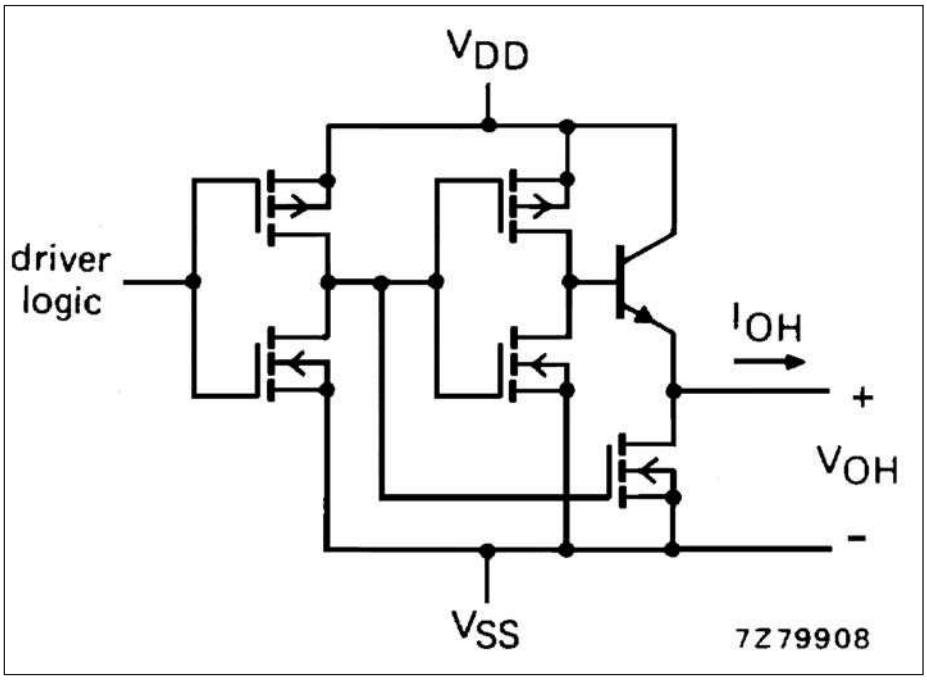


Figura 5: stadio d'uscita BCD to 7-segment Latch/Decoder/Driver 4511.

con un transistor NPN configurato come inseguitore d'emettitore (o a collettore comune), una presenza decisamente inconsueta tra i componenti di tipo CMOS. Le caratteristiche di questo stadio sono quelle di garantire: una notevole amplificazione di corrente (intrinseca dei transistor); un'amplificazione di tensione pressoché uguale a 1 (leggermente minore di 1), ragione per la quale lo stadio è detto anche inseguitore di tensione (*voltage follower*); un'elevata impedenza di ingresso; una bassa resistenza d'uscita.
Certamente questa è la situazione ideale per non caricare (cioè assorbire corrente da) il circuito CMOS che lo pilota e per garantire le migliori condizioni d'uscita, assicurando, a livello alto (1 logico), l'erogazione fino a 25 mA, più che sufficienti per pilotare direttamente il carico (ciascuno dei segmenti-LED del digit). La **figura 5** (di proprietà Philips Semiconductors, estratta dal datasheet del suo HEF4511B) mostra lo schema di questo stadio d'uscita evidenziando la presenza del transistor appena descritto. Su questo settore agisce l'influenza del segnale d'ingresso LT (*Lamp Test, pin3*), attivo basso. Se si forza uno 0 su LT, tutte le linee d'uscita di segmento sono forzate a 1, causando l'accensione di tutti i segmenti del digit, al fine di verificarne l'efficienza. Da notare che questo ingresso è prioritario rispetto a tutti gli altri, compresi LE e BI.

L'analisi della tabella di verità (**figura 6**, *Function Table*) è, come al solito, un importante esercizio, capace di sintetizzare il funzionamento del dispositivo:

a) l'ultima riga evidenzia la forza prioritaria di LT: quando esso è attivo (basso, low level), tutte le uscite di segmento passano a 1 indipendentemente dal valore di ogni altro ingresso, accendendo di fatto il digit (a catodo comune) collegato al decoder;
b) la penultima riga conferma che, dopo LT, comanda BI: se questo ingresso è attivo (basso, L), con LT disattivo, tutte le uscite di segmento passano a 0 (basse, L) spegnendo di fatto il digit;
c) la prima riga mostra l'effetto di LE: se esso è attivo (basso, L), la memoria è trasparente (la parola DCBA d'ingresso passa sul decoder senza alcun intoppo), ma nel passaggio da 0 a 1 (fronte di salita) e per tutto il tempo in cui rimane a 1 la memoria interviene e blocca sulle sue uscite il valore dell'ingresso corrente.

Le caratteristiche logiche di questo componente sono riassunte in dettaglio dallo schema funzionale in **figura 7** e sinteticamente dallo schema pratico in **figura 8**: le linee d'ingresso possono essere quelle del Bus Dati di un microprocessore o le uscite di una porta di un single-chip o della porta parallela di un PC, mentre le linee d'uscita

Figura 6: tabella di verità BCD to 7-segment Latch/Decoder/Driver 4511.

	INGRESSI							USCITE (ATTIVE ALTE)						
n°	$\overline{LE}$	$\overline{BI}$	$\overline{LT}$	D	C	B	A	a	b	c	d	e	f	g
dato	H	H	H	X	X	X	X	dato *memorizzato sul fronte di salita di* $\overline{LE}$						
0	L	H	H	L	L	L	L	H	H	H	H	H	H	L
1	L	H	H	L	L	L	H	L	H	H	L	L	L	L
2	L	H	H	L	L	H	L	H	H	L	H	H	L	H
3	L	H	H	L	L	H	H	H	H	H	H	L	L	H
4	L	H	H	L	H	L	L	L	H	H	L	L	H	H
5	L	H	H	L	H	L	H	H	L	H	H	L	H	H
6	L	H	H	L	H	H	L	L	L	H	H	H	H	H
7	L	H	H	L	H	H	H	H	H	H	L	L	L	L
8	L	H	H	H	L	L	L	H	H	H	H	H	H	H
9	L	H	H	H	L	L	H	H	H	H	L	L	H	H
10	L	H	H	H	L	H	L	L	L	L	L	L	L	L
11	L	H	H	H	L	H	H	L	L	L	L	L	L	L
12	L	H	H	H	H	L	L	L	L	L	L	L	L	L
13	L	H	H	H	H	L	H	L	L	L	L	L	L	L
14	L	H	H	H	H	H	L	L	L	L	L	L	L	L
15	L	H	H	H	H	H	H	L	L	L	L	L	L	L
$\overline{BI}$	X	L	H	X	X	X	X	L	L	L	L	L	L	L
LT	X	X	L	X	X	X	X	H	H	H	H	H	H	H

H livello logico alto
L livello logico basso
X livello logico irrilevante

Figura 6: tabella di verità BCD to 7-segment Latch/Decoder/Driver 4511.

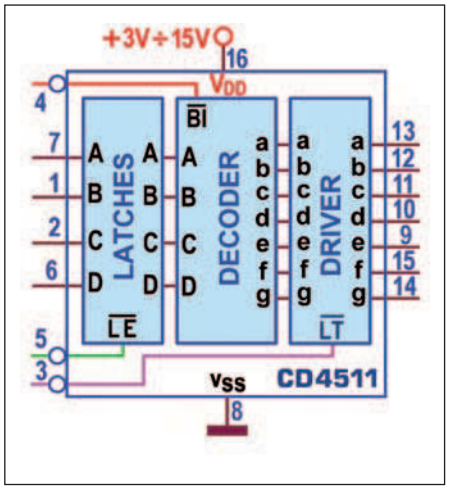


Figura 7: schema funzionale BCD to 7-segment Latch/Decoder/Driver 4511.

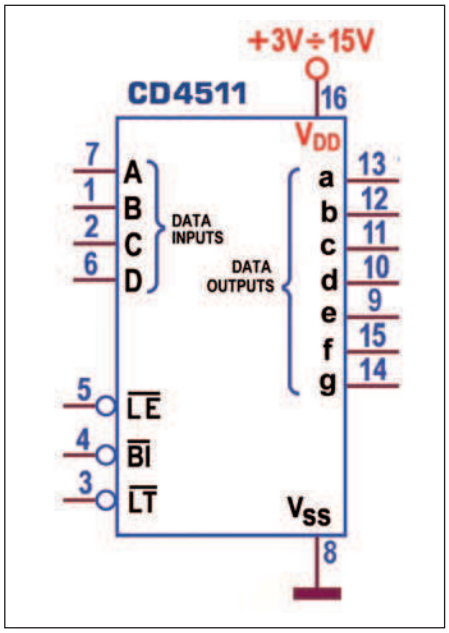


Figura 8: schema pratico BCD to 7-segment Latch/Decoder/Driver 4511.

rendono disponibili i sette valori logici attivi alti necessari per controllare i 7 segmenti di un digit a catodo comune.
Per le scelte di progetto non va dimenticato che sulle uscite di segmento sono presenti valori V_{OH} (tensione d'uscita a livello alto) tipici CMOS, tipicamente tendenti a quello dell'alimentazione V_{DD}; collegando su esse un normale digit a catodo comune è consigliabile la presenza di resistori in serie ai segmenti-LED, necessari per limitare la corrente a valori tali da garantirne una buona accensione in sicurezza. Il valore della loro resistenza dipende dalla tensione d'alimentazione e dalla corrente erogata. Consultando il datasheet è possibile ricostruire una interessante tabella (**figura 9**) relativa alle uscite in condizioni di funzionamento tipiche. Con tensione d'alimentazione di +5V, supponendo di imporre una corrente erogata di 15 mA, risulta disponibile una tensione V_{OH} di 4V che, al netto della caduta di 1,8V sul LED, porta a un valore di resistenza pari a 147ohm, normalizzato a quello standard di 150ohm. La **figura 10** mostra l'impiego del decoder 4511 con un normale digit a catodo comune nelle condizioni appena suggerite; nel normale funzionamento, il piedino di controllo LE (*pin5*) della memoria deve essere posto a massa, mentre i rimanenti due, BI (pin4) e LT (*pin3*), possono essere lasciati inattivi, collegandoli al positivo dell'alimentazione, data la natura CMOS del componente che non ammette pin fluttuanti.
Come per gli altri componenti della famiglia CMOS, la potenza dissipata è trascurabile mentre il ritardo di propagazione (Propagation Delay Time, t_{PHL} e t_{PLH}), rilevato con carico di 50 pF, mediamente si dimezza a ogni incremento di 5V della tensione di alimentazione V_{DD}. Per un componente così sofisticato i punti di riferimento sono numerosi: per esempio per disporre di uscite stabili, dopo l'applicazione di un codice in ingresso, sono necessari da 150ns (V_{DD}=5V) a 40ns (V_{DD}=15V); nei confronti delle uscite dopo l'applicazione del segnale LE è necessario lo stesso tempo, mentre ne basta la metà dopo l'azione di LT.
Ci sono altri integrati della serie CMOS che offrono la disponibilità di decoder da BCD a 7 segmenti; i 4055, 4056 e 4543 sono *Latch/Decoder/Driver* espressamente progettati per visualizzatori a cristalli liquidi, LCD, come il 4544, che ha la possibilità di invertire la logica delle combinazioni d'uscita; il 4547 è un decoder/driver per digit a catodo comune o per lampade a incandescenza, senza memoria e senza LT, ma in grado di garantire una corrente d'uscita fino a 65mA.

Figura 9: dati di progetto BCD to 7-segment Latch/Decoder/Driver 4511.

V_{CC}	I_{OH}	V_{OH}	R (ohm)
5V	0mA	4,40V	
	5mA	4,20V	480
	10mA	4,05V	225
	15mA	4,00V	147
	20mA	3,80V	100
	25mA	3,70V	76
10V	0mA	9,40V	
	5mA	9,20V	1,48k
	10mA	9,10V	730
	15mA	9,00V	167
	20mA	9,00V	360
	25mA	8,90V	284
15V	0mA	14,40V	
	5mA	14,20V	2,48k
	10mA	14,10V	1,23k
	15mA	14,00V	813
	20mA	14,00V	610
	25mA	14,00V	488

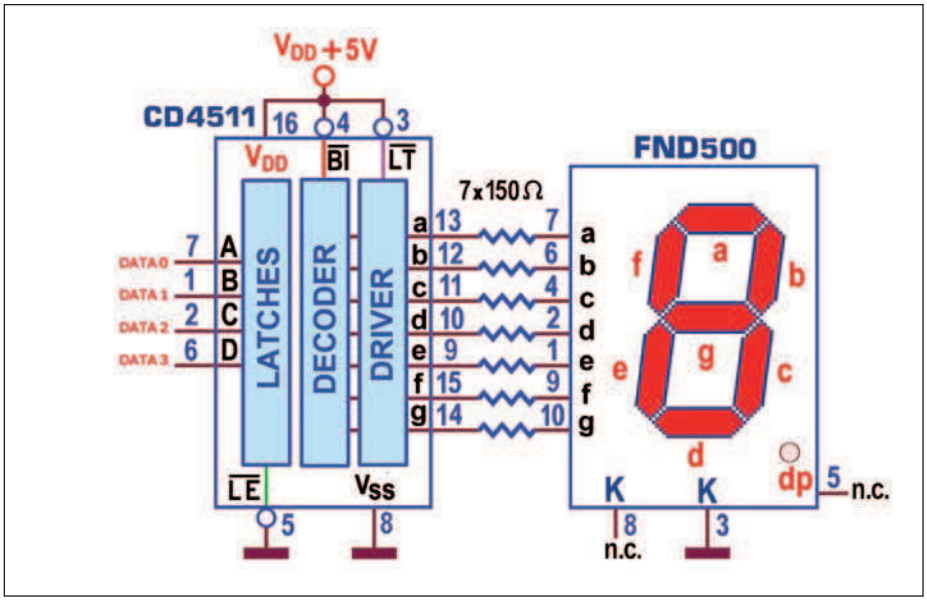


Figura 10: progetto con FND500, BCD to 7-segment Latch/Decoder/Driver 4511.

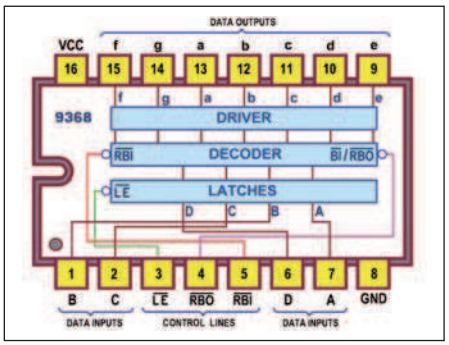


Figura 11: pinout Binary to 7-segment Latch/Decoder/Driver 9368.

Il 4558 e il 4734 (decoder/driver per digit a catodo comune) sono in grado di gestire (unici nella serie CMOS) anche il *Ripple Blanking* per la soppressione degli zero non significativi; il secondo dispone anche di LT e di memoria per cui, per ospitare RBO e RBI, fruisce di un inconsueto contenitore a 18 piedini; i 4026 e 4033 sono contatori/divisori decadici con uscite decodificate adatte a display a 7 segmenti.

Un cenno a parte merita il 4495, un Latch/Decoder/Driver del tutto anomalo rispetto ai precedenti, non solo per la strana disposizione dei piedini associati alle 7 uscite di segmento e ai 4 ingressi di dato (in tutti gli altri sempre nelle stesse posizioni del 4511), ma anche per capacità di assicurare la decodifica di tutte le 16 combinazioni del codice binario a 4 bit (come il 9368, trattato tra poco), ma in modo decisamente originale, utilizzando una ROM 16x9 bit programmabile (per cui la Motorola, che lo produce, offre su richiesta altre possibili diverse programmazioni); poiché contiene internamente un resistore da 290 ohm su ogni linea d'uscita, alimentandolo a 5V per pilotare un digit a LED può essere utilizzato senza resistori esterni.

DECODER DA BINARIO A 7 SEGMENTI (TTL): 9368

Questa lunga rassegna termina con il decoder TTL 9368, un componente eccezionale, unico in grado di decodificare tutte le 16 combinazioni del codice binario puro a 4 bit fornito sui suoi ingressi; sebbene il suo datasheet non preveda per esso una classificazione specifica, lo chiameremo Binary to 7-segment Decoder, in contrapposizione con quelli finora trattati, tutti appartenenti alla categoria destinata a decodificare solo un codice BCD.

Esso è sostanzialmente pin-out compatibile con il TTL 74LS48 (esclusa la funzione associata al *pin3*) e funzionalmente simile al CMOS 4511, per altro (come detto) entrambi destinati a decodificare solo 10 codici su 16. Il suo pin-out è illustrato in **figura 11**. La figura evidenzia la stessa struttura logica del CMOS 4511, in grado di assicurare le sue stesse tre mansioni in un unico componente: lo strato di memoria a 4 bit sulle linee d'ingresso, la funzione di decoder (ora estesa a tutte e 16 le possibili combinazioni) e la disponibilità di driver su ciascuna delle 7 linee d'uscita. Va subito anticipato che le descrizioni relative al funzionamento del 9368 sono in gran parte simili a quelle fornite per il 4511, fatta salva la puntualizzazione di alcuni piccoli particolari. Di certo questo straordinario componente ne migliora i servizi in comune, rendendo possibile (tra l'altro) la tecnica Ripple *Blanking*, che riprenderemo in dettaglio tra poco.

Lo strato di memoria

Lo strato di memoria e il suo controllo è identico a quello già descritto, tenendo presente che ora il segnale di sincronismo LE, *Latch Enable*, è sul *pin3*. In sintesi, ciascuna delle 4 linee d'ingresso attraversa un flip-flop di tipo *D-Latch* e il codice a esse relativo può essere congelato a valle degli elementi di memoria sul fronte di salita di LE e mantenuto costante (in ingresso al decoder) fino a quando la linea LE verrà riportata a livello 0; per tutto il tempo nel quale LE è a livello 0 il codice binario puro a 4 bit applicato in ingresso passa inalterato a valle della memoria, come se essa non ci fosse.

Naturalmente le applicazioni suggerite per il 4511 (intertempo per cronometro e gestione di frequenzimetro) sono assolutamente legittime anche con il 9368, ora però da associare a contatori TTL come il *Decade Counter 74LS90 o il Binary Counter 74LS93*: se l'ingresso LE è mantenuto a massa, il digit a esso collegato mostrerà le corrispondenti cifre (rispettivamente decimali o esadecimali) variabili con la stessa cadenza imposta dalla frequenza di clock applicata al contatore; non appena LE viene scollegato da massa il simbolo sul digit, corrispondente al codice presente in quel momento sugli ingressi DCBA, non subirà più aggiornamenti (sebbene il conteggio fornito in ingresso continui a variare regolarmente), rimanendo invariato fino a quando LE sarà riportato a massa.

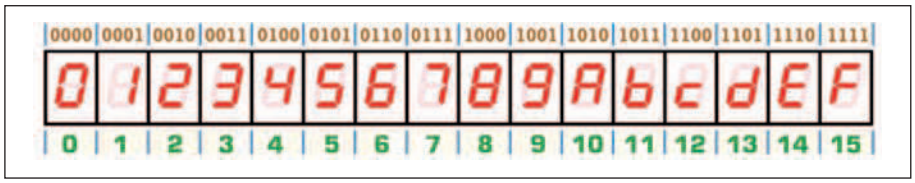


Figura 12: simboli visualizzati con FND500 Binary to 7-segment Latch/Decoder/Driver 9368.

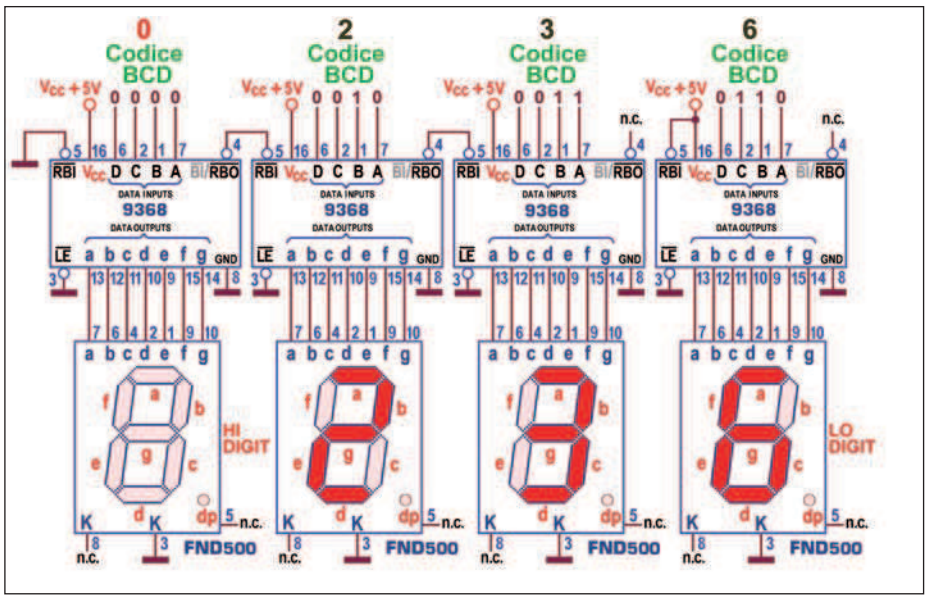


Figura 13: effetto Ripple Blanking, visualizzatore a 4 digit FND500 con 9368.

La funzione di decoder

Unico tra tutti i decoder conosciuti, questo componente interpreta tutte le possibili 16 combinazioni del codice binario a 4 bit proposto in ingresso, assicurando per ciascuna di esse il corrispondente simbolo del sistema di numerazione esadecimale; l'aspetto del digit collegato al decoder è dunque quello mostrato in **figura 12**. La grande novità consiste nella visualizzazione dei simboli alfabetici da A a F (proposte con lettere minuscole o maiuscole) in aggiunta ai simboli numerici da 0 a 9 (già disponibili con gli altri decoder); tra l'altro nella visualizzazione del 6 e del 9 è previsto il famoso trattino, rispettivamente in alto o in basso. Su questa parte circuitale agisce l'influenza dei segnali BI/RBO (*Blanking Input/Ripple Blanking Output*, pin4) e RBI (*Ripple Blanking* Input, *pin5*), entrambi attivi bassi, gli stessi descritti la puntata precedente per il decoder TTL 74LS48 che, come già anticipato, da questo punto di vista è del tutto identico a quello sotto test. Ne ricordiamo le principali considerazioni.

In virtù di una particolare circuiteria interna (basata su un resistore di pull-up, adatta a realizzare una logica di tipo *wired-AND* con ogni uscita open *collector*) la linea collegata al pin4 può essere utilizzata, senza produrre danni, sia come uscita (rendendo disponibile il segnale RBO) sia come ingresso (consentendo di forzare su di essa il segnale BI). Usata come ingresso BI permette (se attivata, cioè forzata a 0) di spegnere incondizionatamente il digit collegato al decoder, con qualunque codice presente sugli ingressi DCBA; nel funzionamento normale BI deve essere lasciato scollegato (o collegato al +5V). Da notare che (come per il 74LS48) l'ingresso BI è prioritario, cioè quando esso è attivo contemporaneamente a ogni altro ingresso la funzione di questi ultimi viene ignorata. Anche in questo caso, applicando un'onda quadra non simmetrica sull'ingresso BI e modificandone il duty cycle, è possibile controllare l'intensità luminosa del digit a esso collegato, come descritto in precedenza. Usata come uscita, la linea RBO serve (insieme all'ingresso RBI sul *pin5*) per gestire il meccanismo di *Ripple Blanking* atto a evitare di accendere tutti gli zero non significativi nell'ambito di un visualizzatore decimale con numerose cifre, sia a sinistra che a destra (se il numero mostrato è frazionario). Sebbene esso sia stato già descritto la puntata scorsa, per il 74LS48, è bene riprenderne i concetti, anche per il fatto che gli schemi applicativi sono diversi per l'assenza della linea di *Lamp Test* (LT) ora sostituita da quella di *Latch Enable* (LE):

a) l'ingresso RBI consente (se attivato, cioè forzato a 0) di spegnere (*blanking*) il digit collegato al decoder solo se contemporaneamente il codice presente sui suoi ingressi DCBA è $(0000)_2$; in queste condizioni forza a 0 anche l'uscita RBO. Se il codice presente sugli ingressi DCBA è diverso da $(0000)_2$, il digit mostra regolarmente i simboli corrispondenti e l'uscita RBO rimane disattiva (cioè a livello logico 1). La condizione per poter visualizzare comunque l'eventuale cifra zero è di lasciare l'ingresso RBI scollegato (o meglio collegato al +5V), cioè disattivo (forzato a livello logico 1);

b) l'uscita RBO serve per reiterare questo servizio anche sul decoder di un eventuale digit meno significativo, posto a destra del primo. Collegato all'ingresso RBI del decoder successivo produrrà lo stesso effetto, tenendo spento il digit a esso collegato solo se sui suoi ingressi DCBA è presente il codice $(0000)_2$;

c) collegando in cascata le uscite RBO di ciascun decoder (a cominciare da quello associato al digit più significativo) con l'ingresso RBI del decoder successivo (subito a destra) è dunque possibile evitare di accende-

re tutti gli antiestetici e inutili zero a sinistra di un numero. Anche nella pratica comune sembra logico scrivere 236 piuttosto che 0236.

Va per altro notato che se il 9368 è utilizzato nel pieno della sua specificità (quella di mostrare numeri esadecimali) questo servizio è comunque illogico, sia per il fatto che ogni cifra del numero mantiene la sua necessità d'esistere (anche gli zero a sinistra, basti pensare a un indirizzo o un dato a 16 bit, come $0AF5_H$ o come $001E_H$), sia per l'improbabilità di dover mostrare un numero esadecimale frazionario (rendendo così inutile anche il servizio per gli zero a destra). Ma è pur vero che questo dispositivo si comporta benissimo anche nell'ambito di un visualizzatore decimale, dando in questo caso per scontato che l'informazione fornita in ingresso ai decoder sia di tipo BCD, anche se essi sarebbero in grado di processare tutte le 16 possibili combinazioni. La situazione è mostrata in **figura 13**, relativa a un visualizzatore con 4 digit FND500, ciascuno controllato da un 9368, nell'ipotesi che i codici BCD in ingresso ai 4 decoder siano quelli corrispondenti alle cifre 0236; con questa predisposizione (ingresso RBI del decoder collegato al digit più significativo forzato a massa) lo "0" corrispondente al codice $(0000)_2$, fornito sui suoi ingressi DCBA, non viene mostrato (il digit rimane spento). Questa situazione forza a 0 anche l'uscita RBO che, a sua volta, forza a massa l'ingresso RBI del decoder collegato al digit successivo (subito a destra). Poiché il codice presente sugli ingressi DCBA di questo decoder non è $(0000)_2$, il fatto che il suo ingresso RBI sia a livello logico 0 è ora del tutto irrilevante e, da questa posizione in poi ogni cifra verrà visualizzata, anche gli eventuali "0" interni. In caso contrario anche il secondo digit rimarrebbe spento e, in coerenza con la regola, il decoder a esso associato forzerebbe a 0 la sua uscita RBO e, di conseguenza, anche l'ingresso RBI del decoder collegato al digit successivo (il terzo da sinistra); il meccanismo si applica per tutte le cifre del display e, se fosse mantenuto inalterato per tutti i decoder successivi al primo, fornendo 4 codici

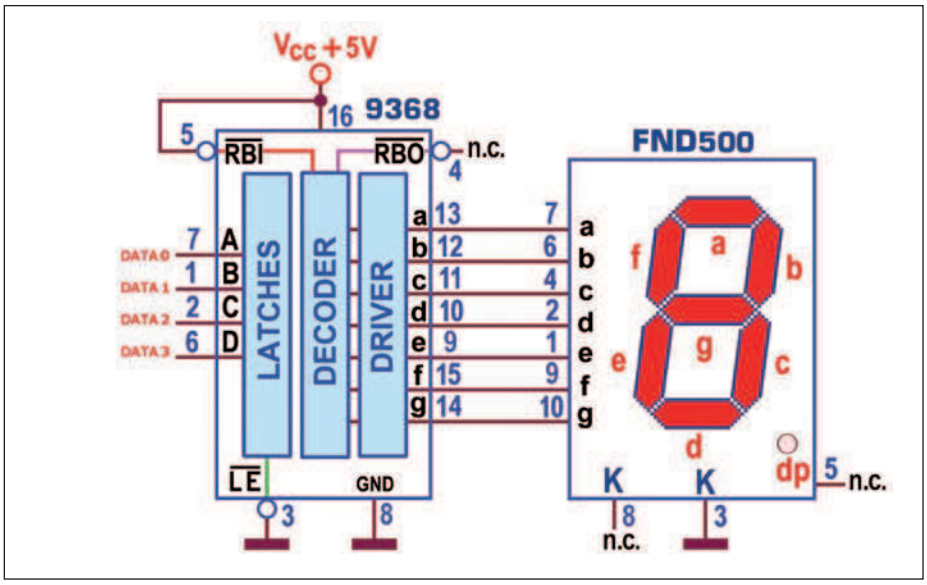


Figura 14: progetto con FND500, Binary to 7-segment Latch/Decoder/Driver 9368.

Figura 15: tabella di verità Binary to 7-segment Latch/Decoder/Driver 9368.

	INGRESSI							USCITE						
n°	$\overline{LE}$	$\overline{RBI}$	D	C	B	A	$\overline{RBO}$	a	b	c	d	e	f	g
dato	H	X	X	X	X	X	H	dato *memorizzato sul fronte di salita di* LE						
0	L	L	L	L	L	L	L	L	L	L	L	L	L	L
		H	L	L	L	L	H	H	H	H	H	H	H	L
1	L	X	L	L	L	H	H	L	H	H	L	L	L	L
2	L	X	L	L	H	L	H	H	H	L	H	H	L	H
3	L	X	L	L	H	H	H	H	H	H	H	L	L	H
4	L	X	L	H	L	L	H	L	H	H	L	L	H	H
5	L	X	L	H	L	H	H	H	L	H	H	L	H	H
6	L	X	L	H	H	L	H	H	L	H	H	H	H	H
7	L	X	L	H	H	H	H	H	H	H	L	L	L	L
8	L	X	H	L	L	L	H	H	H	H	H	H	H	H
9	L	X	H	L	L	H	H	H	H	H	H	L	H	H
10	L	X	H	L	H	L	H	H	H	H	L	H	H	H
11	L	X	H	L	H	H	H	L	L	H	H	H	H	H
12	L	X	H	H	L	L	H	H	L	L	H	H	H	L
13	L	X	H	H	L	H	H	L	H	H	H	H	L	H
14	L	X	H	H	H	L	H	H	L	L	H	H	H	H
15	L	X	H	H	H	H	H	H	L	L	L	H	H	H
$\overline{BI}$	X	X	X	X	X	X	L	L	L	L	L	L	L	L

H livello logico alto
L livello logico basso
X livello logico irrilevante

BCD "nulli", terrebbe spento tutto il visualizzatore. Sebbene la cosa sia possibile non sembra comunque una buona soluzione: se sugli ingressi DCBA di ogni decoder viene fornito il codice $(0000)_2$ è ragionevole mostrare almeno lo zero associato al digit più a destra; per questa ragione l'ingresso RBI del decoder a esso relativo viene mantenuto a 1, spezzando la catena di controllo e vanificandone l'effetto.
Nella scorsa puntata abbiamo analizzato anche lo schema necessario per gestire gli zero non significativi della parte frazionaria, per esempio come nel numero 63,20; poiché la tecnica rimane sostanzialmente identica evitiamo di appesantire ulteriormente questa trattazione, rimandando il lettore a quanto già pubblicato.

Lo strato del driver

La terza parte di questo componente è decisamente interessante ed esclusiva: ciascuna delle uscite interne del decoder è dotata potente driver in grado di erogare verso i segmenti-LED una corrente I_{OH} da 16mA (minima) a 22mA (massima), senza bisogno di resistori per limitare la corrente; la tensione nominale d'uscita a livello alto (V_{OH}) è 3,4V (secondo il datasheet), ma con un LED a essa direttamente collegato si adatterà al valore della caduta ai suoi capi, cioè da 1,75V a 1,85V.
Nell'impiego con un normale digit a catodo comune la presenza di resistori di limitazione corrente in serie ai segmenti-LED non è dunque necessaria. La **figura 14** propone questa situazione, sottolineando che nel normale funzionamento, il piedino di controllo LE (*pin3*) della memoria *latch* deve essere a massa mentre le due linee di controllo del *Ripple Blanking*, RBO (pin4) e RBI (*pin5*), irrilevanti in questo caso, possono essere lasciate inattive, lasciando scollegata la prima e scollegata (o meglio collegandola a +5V) la seconda.
Anche in questo caso la tabella di verità (*Function Table*, **figura 15**) offre un'eccellente sintesi del funzionamento del dispositivo, per cui è importante saperla consultare e capire:
a) l'ultima riga evidenzia la forza prioritaria di BI, sul pin4 usato come ingresso: quando esso è attivo (basso, L), tutte le uscite di segmento passano a 0 indipendentemente dal valore di ogni altro ingresso, spegnendo di fatto il digit collegato al decoder;

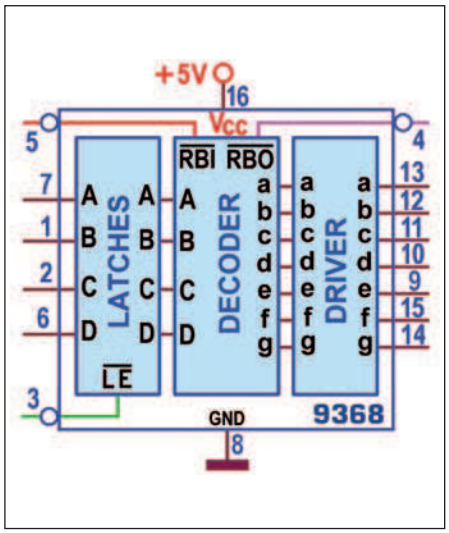


Figura 16: schema funzionale Binary to 7-segment Latch/Decoder/Driver 9368.

b) la prima riga mostra l'effetto di LE: se esso è tenuto a 1 (alto, H) la memoria mantiene sulle uscite il valore presente sugli ingressi DCBA nel suo passaggio da 0 a 1 (fronte di salita); riportata a 0 la memoria è trasparente (la parola DCBA d'ingresso passa sul decoder senza alcun intoppo);
c) la seconda riga conferma l'influenza dell'ingresso RBI sul simbolo mostrato sul digit e su RBO (pin4 usato come uscita): se forzata a 0 (attiva bassa, L), essa produce effetto solo quando sugli ingressi DCBA è presente il codice $(0000)_2$ ed è irrilevante in tutti gli altri casi.

Le caratteristiche logiche di questo componente sono riassunte in dettaglio dallo schema funzionale (**figura 16**) e sinteticamente dallo schema pratico (**figura 17**).
In conclusione, l'utilizzo del 9368, prodotto dalla Fairchild Semiconductor da più di vent'anni, è molto versatile in ogni applicazione, ma risulta particolarmente interessante quando è chiamato a operare nell'ambito dei microprocessori o dei microcontrollori.Essendo l'unico decoder in grado di organizzare la visualizzazione dei simboli esadecimali, indispensabili per tradurre in modo umano le comunicazioni binarie, si presta in modo eccellente per visualizzare le informazioni tipicamente presenti sulle porte d'uscita di un micro o di un PC, come indirizzi (delle locazioni di memoria, visualizzabili su 4 digit a partire dai loro 16 bit) o dati (per esempio il contenuto di registri della CPU o di una locazione

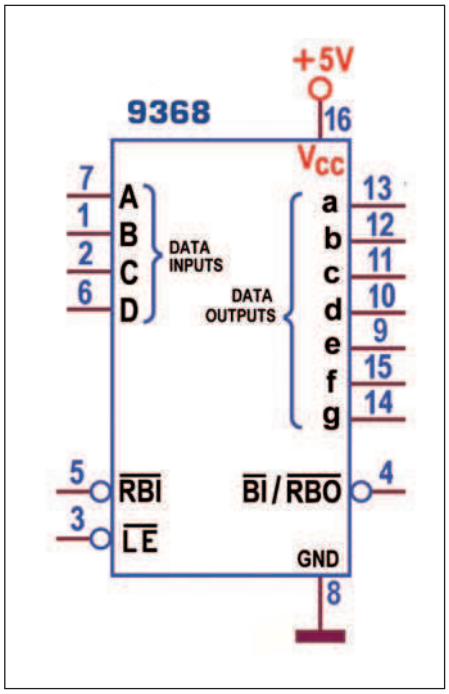


Figura 17: schema pratico Binary to 7-segment Latch/Decoder/Driver 9368.

della memoria, visualizzabili su 2 digit a partire dai loro 8 bit). La sua disponibilità di un stato di memoria rende inoltre possibile l'interfacciamento diretto con le linee del Bus Dati di un processore o di un single-chip, controllando il sincronismo di memorizzazione (LE) a partire dai Bus Address e Bus di Controllo, con opportune reti di decodifica binaria.
L'assorbimento tipico è di 335 mW e il ritardo di propagazione (Propagation Delay Time, rilevato con carico di 100ohm/15pF) per disporre di uscite stabili è pari t_{PLH}=75ns e t_{PLH}= 50ns, dopo l'applicazione di un codice in ingresso, e leggermente maggiore dopo l'applicazione del segnale LE.
Per finire possiamo citare altri due componenti, del tutto pinout compatibili con il 9368 ma dotati di uscite attive basse, adatte quindi a digit ad anodo comune, come l'FND507: il 9370, versione con uscite open collector sempre prodotta dalla Fairchild Semiconductor, funzionalmente identico e capace di gestire tutti e 16 i codici a 4 bit proposti in ingresso, ma che ha bisogno di resistori esterni; e il 8374, prodotto dalla NTE Electronics, capace di assicurare 15mA in uscita ma limitato a interpretare solo ingressi di tipo BCD. ❑

CODICE MIP 2757075

(parte quarta)

INTRODUZIONE AL VOIP

In questa quarta e ultima puntata affronteremo insieme il protocollo SIP. Esso rappresenta la tendenza più attuale nella progettazione di sistemi VoIP

Nel 1999 un gruppo di ingegneri facenti capo allo IETF (Internet Enginering Task Force) mise a punto la prima versione del protocollo SIP, acronimo di *Session Initiation Protocol*, rilasciando il primo documento ufficiale battezzato come RCF 2543. La sigla RCF non ha nessuna attinenza con marchi commerciali, essa significa semplicemente *Request for Comments* e i lettori più attenti ricorderanno che è già stata incontrata nella seconda puntata. L'obiettivo del loro lavoro, portato a termine quando esisteva già un sistema valido come l'H323 (vedi terza puntata), era quello di fornire un protocollo di facile applicazione, ottimo anche per uso domestico, abbastanza indipendente dall'hardware e dal protocollo di trasporto utilizzato. Facendo un doveroso paragone, si può dire che chi progettò l'H323 lo fece con una prospettiva quasi esclusivamente telefonica. Il SIP, d'altro canto, è stato pensato in un momento in cui appariva necessaria la convergenza di tutti i servizi informatici come e-mail, database, applicazioni aziendali e naturalmente VoIP. Possiamo anche pensare al SIP come a uno standard aperto a qualsiasi futura implementazione, privo di complicate strutture protocollari e facilmente utilizzabile in ogni architettura preesistente. Un buon punto di forza del SIP è nel somigliare molto ad altri protocolli decisamente familiari a chi si occupa di Internet: il protocollo HTTP e il protocollo SMPT, quest'ultimo usato per la posta elettronica. Attualmente lo IETF fa riferimento alla raccomandazione RCF 3261 e seguenti (RCF 3262, RCF 3903, ecc.) per individuare le caratteristiche del protocollo SIP. Nel corso degli anni si è assistito a un progressivo avvicinamento delle funzionalità fra H323 e SIP, tuttavia è bene sottolineare che sono protocolli progettati con diverse filosofie d'impiego e le differenze concettuali sono molte.

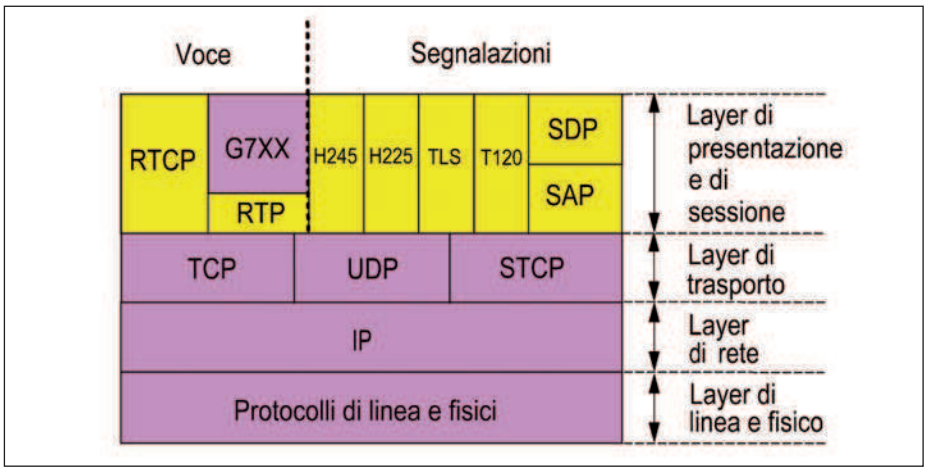


Figura 1: suite protocollare SIP all'interno dello stack ISO-OSI.

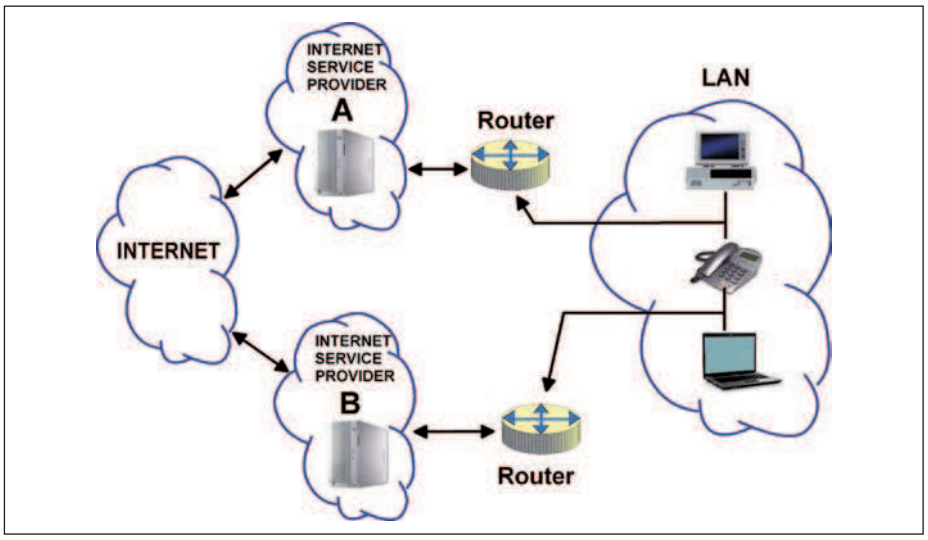


Figura 2: esempio di multihoming fra una LAN e rete Internet.

ARCHITETTURA PROTOCOLLARE DEL SIP

Anche la suite SIP fa uso di protocolli ampiamente utilizzati e quindi di sicura affidabilità. Alcuni di essi li abbiamo già incontrati nella terza puntata quando parlavamo dello standard H323. **In figura 1** è possibile osservare lo stack protocollare completo (i protocolli che compongono lo standard SIP sono colorati in giallo). Possiamo inoltre notare i protocolli essenzialmente destinati alla voce, distinti dagli standard dedicati invece alle segnalazioni e alla sicurezza delle informazioni. Alcuni protocolli operanti al livello di layer di pre-

sentazione e sessione, come SDP e SAP, non sono mai stati affrontati nella nostra trattazione. Niente paura, lo faremo in seguito. Anche a livello del layer di trasporto, può essere usato un protocollo che non abbiamo mai incontrato prima: l'SCTP (*Stream Control Transmission Protocol*). Esso può essere "scelto" in una suite comprendente i ben noti UDP e TCP, protocolli già ampiamente trattati nelle puntate scorse. L'STCP è un protocollo molto versatile, sicuramente destinato a sostituire il TCP da qui a breve. Nato con la sigla MDTP, questo protocollo cambierà nome in SCTP solo dopo nove versioni. La lunga gestazione si rese necessaria per progettare un sistema di connessione veramente sicuro e dotato di peculiarità destinate a non farlo invecchiare troppo precocemente. Vediamo dunque le caratteristiche fondamentali di questo protocollo:

- possibilità di connessioni multiple;
- alta tolleranza ai guasti;
- la consegna dei dati non è legata a una sequenza particolare;
- possibilità di implementare *load balancing*.

Esaminiamole una ad una. La possibilità di connessioni multiple è un opzione molto importante, quasi inopinabile dal momento che la tendenza attuale nella progettazione di reti telematiche è quella di offrire all'utente una diversificazione nella scelta dei protocolli usati nel collegamento. Non solo, è possibile stabilire collegamenti con più indirizzi e addirittura testare, per mezzo di appositi pacchetti, la presenza in rete degli indirizzi prescelti. In questo modo il collegamento non verrà abbattuto se uno dei nodi non è più in attività o se l'indirizzo viene cambiato. Sarà infatti sufficiente utilizzare un altro nodo, e quindi un altro percorso, per mantenere il collegamento. Per questa peculiarità, l'SCTP viene definito un protocollo orientato all'associazione e non al collegamento. Le potenzialità di indirizzamento multiplo, definite anche *multihoming*, consentono inoltre un'immunità ai guasti non supportabile dagli altri protocolli fin qui studiati. I concetti fin qui espressi sono riassunti in **figura 2**, dove il *multihoming* è rappresentato in un segmento di rete piuttosto classico: una rete locale (LAN) collegata a Internet a mezzo di provider.

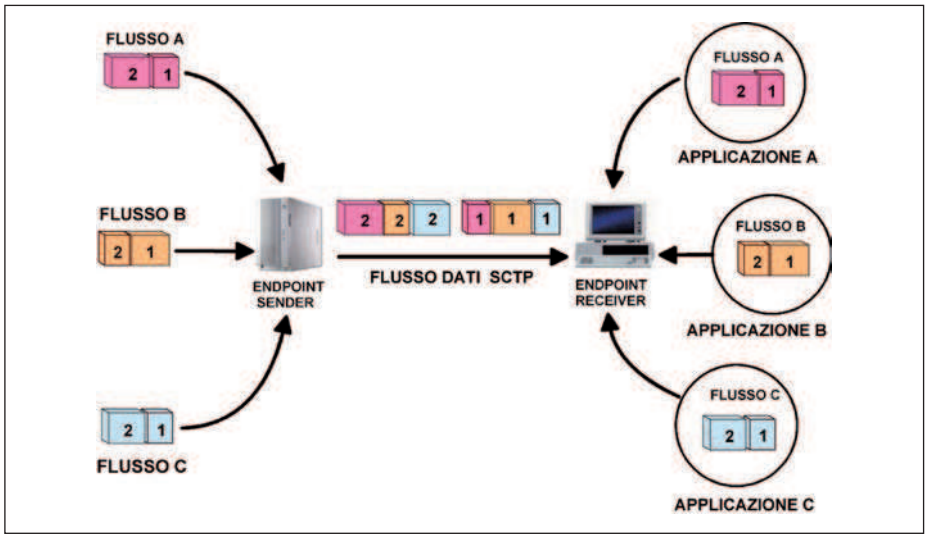


Figura 3: multiplexer associativo mediante protocollo SCTP.

Un'altra caratteristica importante è costituita dal poter disporre di datagrammi (pacchetti) che trasportano dati relativi ad applicazioni diverse. In questo modo si realizza un multiplexer virtuale, ovvero realizzabile senza ulteriori implementazioni hardware, in cui diverse applicazioni possono partecipare allo scambio dati fra due entità per mezzo di un'associazione SCTP. I dati, poi, possono essere inviati anche senza un ordine predefinito. Infatti, i singoli *payload* sono comunque associati a un identificatore che può essere utilizzato dall'applicazione per ricostruire l'intera trama. In **figura 3** è schematizzato questo concetto (nell'esempio mostrato i dati sono trasmessi in sequenza) che viene spesso definito con il nome di multistream.

Il *load balancing,* altra caratteristica importante dell'SCTP, consiste in una serie di precauzioni destinate a non sovraccaricare la larghezza di banda del mezzo trasmissivo e, conseguentemente, evitare perdite di dati. Ciò è ottenuto con alcune metodologie quali, ad esempio, "consigliare" all'ente ricevente un buffer minimo per i dati trasmessi oppure cominciare a trasmettere piccole quantità di dati per poi aumentare progressivamente. In questo caso la perdita di pacchetti definirà una soglia massima di capacità ricettiva che non dovrà essere superata nella sessione. Il formato dei datagrammi SCTP è mostrato in **figura 4**.

Come possiamo notare vi è un *header*, elemento presente in tutti i protocolli che finora abbiamo affrontato, seguito da un numero variabile di *chunk*. La parola *chunk*, letteralmente "pezzetto", è usata per definire parti del datagramma che contengono i dati trasmessi (*payload*) oppure altre informazioni relative alla sessione. Esistono quattordici tipologie di *chunk*, alcune di queste devono necessariamente essere trasmesse singolarmente, ovvero senza altri *chunk* accodati. Tutte le tipologie di *chunk* sono riassunte in **tabella 1** ove sono anche riportati i loro acronimi e le relative descrizioni.

A questo punto si potrebbe pensare che il protocollo SCTP, per la sua completezza e versatilità, sia effettivamente una scelta giusta rispetto a UDP o TCP. Tuttavia, come sempre, la scelta del protocollo è fatta in funzione delle caratteristiche della rete e del contenuto informativo che stiamo trattando (voce, dati, video, ecc.). Tutto questo per dire che il protocollo SCTP, al di là dei suoi tanti pregi, in una sessione VoIP potrebbe non garantire la performance del TCP (e ancor più dell'UDP) in termini di capacità di trasferimento dati nell'unità di tempo. Viceversa, l'occupazione di banda del protocollo SCTP è decisamente modesta se confrontata, ad esempio, con quella del TCP. In **figura 5** e **figura 6** si può vedere l'espressione grafica dei concetti appena accennati.

La penalizzazione del data trasfer dell'SCTP dipende, in primis, dal fatto che le

Tabella 1 – Tipologie di chunk

NUMERO IDENTIFICATIVO	ACRONIMO	TIPOLOGIA	CHUNK AGGIUNTIVI
0	DATA	Dati trasmessi	Si
1	INIT	Inizializzazione collegamento	No
2	INIT ACK	Conferma inizializzazione	No
3	SACK	Controllo del flusso dati e congestione	Si
4	HEARTBEAT	Verifica attività indirizzo IP (per multihoming)	Si
5	HEARTBEAT ACK	Conferma di indirizzo IP attivo	Si
6	ABORT	Abbatte un'associazione SCTP	Si
7	SHUTDOWN	Richiesta di chiusura collegamento	No
8	SHUTDOWN ACK	Conferma di chiusura collegamento	No
9	ERROR	Errore (varie tipologie)	Si
10	COOKIE ECHO	Usati durante l'inizializzazione di una connessione SCTP	Si
11	COOKIE ACK		
12	ECNE	Notifica di congestione traffico	Si
13	CWR	Riduzione della capacità trasmissiva a seguito di congestione per traffico	Si

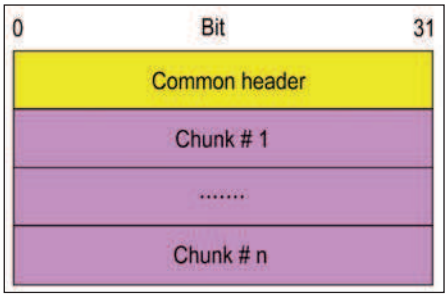


Figura 4: formato generico di un pacchetto SCTP.

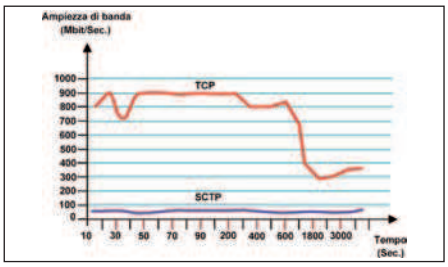


Figura 5: ampiezza di banda (bandwidth) dell'STCP confrontata con TCP.

caratteristiche principali di questo protocollo sono la sua sicurezza e la sua estrema elasticità d'impiego. Cominciamo ora a entrare nel dettaglio delle entità hardware-software definite nel protocollo SIP, così come abbiamo visto, nelle puntate scorse, per il protocollo H323.

SIP USER AGENT

E' la cellula elementare di ogni impianto (acronimo UA) e viene anche definito terminale o end-point. Trattasi di un dispositivo hardware-software in grado di effettuare una sessione di scambio dati/voce, utilizzando il protocollo SIP. In pratica può essere un telefono dedicato che invece di essere connesso alla linea PSTN utilizza un nodo IP, un PC (se lo scambio riguarda solo dati) oppure un PC ove il software in esecuzione permette la simulazione di un telefono. Quest'ultima possibilità, molto sfruttata anche da gestori indipendenti, viene anche indicata con il termine di *soft-phone*. Tra l'altro, in Internet è possibile trovare programmi che simulano un *soft-phone*, da utilizzare con il proprio PC, il cui utilizzo è completamente gratuito. Esistono due tipi di UA, la cui diversificazione è molto semplice. Viene definito UA client (acronimo UAC) l'end-point che inizia la sessione o chiamata. UA server (acronimo UAS) è invece il terminale che riceve la chiamata. Lo stesso UA di una rete SIP può quindi agire sia da client che da server, dipende dalla direzione della chiamata. Naturalmente, nell'ambito della stessa sessione, non potrà esserci uno scambio di ruolo per questo tipo di elemento. Sarà bene, per una completa comprensione del protocollo SIP, familiarizzare subito con il paradigma client-server sopra accennato, ricordando che la differenza fra UAC e UAS è solamente logica (e non, per esempio, hardware).

SIP PROXY SERVER

Il Proxy Server operante in un'infrastruttura SIP è generalmente associato a una ben definita area geografica o zona. Esso è costituito, nella maggior parte dei casi, da un PC su cui gira un programma in grado di gestire le chiamate. Se richiesto, può effettuare una gestione "personalizzata" dei diversi end-point costituenti la sua zona di competenza (rifiuto chiamate per certi numeri, tariffazione, ecc.). Se per motivi vari non è in grado di svolgere le sue funzioni specifiche, esso può chiamare a sua volta un altro proxy a cui demandare le funzionalità necessarie. Questa caratteristica rende il proxy server in grado di funzionare anche da client. Osservando la **figura 7** si noterà che il lavoro principale di questo componente, tuttavia, è l'instradamento (*routing*) delle varie chiamate e risposte provenienti dagli end-point di sua competenza.
Sempre in **figura 7** è possibile osservare schematicamente il concetto di scambio dei ruoli fra UAC e UAS. Questa caratteristica, come si può osservare, dipende dalla direzione delle richieste e delle risposte. Esiste però un'ulteriore classificazione dei Proxy Server, che possono dividersi in:

- stateless;
- *statefulll*.

I proxy stateless sono molto semplici poiché non conservano memoria della transizione (sessione SIP) che effettuano. Per usare una terminologia da addetti ai lavori, si può anche dire che sono completamente "trasparenti" ai vari messaggi del protocollo SIP. E' interessante sottolineare che, nell'ambito di una stessa sessione SIP, lo stesso Proxy Server può assumere entrambe le funzioni (seppure in fasi distinte). I *proxy statefull*, dal funzionamento più complesso, si dividono invece in altre due ulteriori categorie:

- *transaction statefull*;
- *call statefull*.

I *transaction statefull* conservano la traccia soltanto delle richieste e delle risposte dei vari UA (siano essi UAC o UAS). Nessun altra informazione riguardo la sessione SIP in attività viene registrata o memorizzata. I server proxy *call statefull*, viceversa, conservano memoria dell'intera sessione e lasciano tracce evidenti della loro presen-

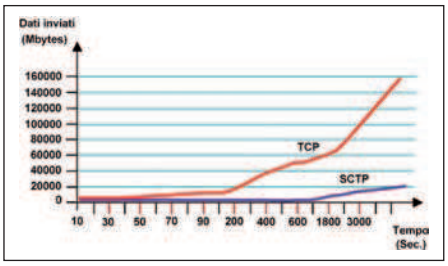


Figura 6: capacità di trasferimento dati (data transfer) dell'SCTP confrontata con TCP.

za nella sintassi dei messaggi del protocollo SIP. La funzione principale dei *call statefull* è la tariffazione del servizio VoIP offerto. Un'altra funzione, molto utile, che può essere svolta da entrambe le tipologie è la cosiddetta replicazione delle chiamate, spesso abbreviata con il termine *forking*. Si tratta, in sostanza, di ripetere per un numero determinato di indirizzi la stessa chiamata. Vedremo fra breve quale può essere il vantaggio nell'utilizzare una simile tecnica. Prima di entrare nel dettaglio occorre però fare la conoscenza della sintassi di un indirizzo SIP. All'inizio di questa puntata abbiamo sottolineato la somiglianza fra il protocollo SIP e altri protocolli sicuramente più noti quali l'HTTP e l'SMPT. A conferma di ciò vi è da dire che tale somiglianza riguarda anche la sintassi degli indirizzi SIP. Ad esempio:

sip:mario-rossi@ditta.magazzino.com

è un indirizzo valido per un'infrastruttura SIP, ma senza il preambolo "sip:" lo sarebbe anche per una casella e-mail di posta elettronica. Un modo più ragionato per interpretare un indirizzo SIP è pensare che l'utente mario-rossi è registrato presso il server ditta.magazzino.com. Quest'ultima dicitura, è bene sottolinearlo, non è il nome del server ma il suo indirizzo, e come tale può essere espresso anche in forma numerica. L'indirizzo del server a cui l'utente appartiene, spesso fornito dal suo gestore di servizi Internet (provider) è anche definito dominio. Supponiamo ora che il nostro Mario Rossi abbia a disposizione più terminali telefonici, uno per ogni zona geografica ove è possibile risieda. In tal caso potremmo avere una situazione di questo tipo:

sip:mario-rossi@ditta.uffici.com
sip:mario-rossi@abitazione.it
sip:mario-rossi@casa.vacanze.de

Il primo indirizzo si riferisce sempre all'azienda ove lavora Mario Rossi, ma in altro locale (uffici, per l'esattezza). Il secondo indirizzo fa evidente riferimento alla sua abitazione, mentre il terzo possiamo dedurre sia l'indirizzo telefonico della sua casa per le vacanze. Con la funzione di *forking* sopra accennata sarà possibile chiamare contemporaneamente tutti gli end-point che fanno riferimento a Mario Rossi. Quest'ultimo si presuppone risponderà solo presso il terminale interessato. Questo esempio ci fa capire quante funzionalità possa offrire un sistema di telecomunicazioni basato su IP.

Tabella 2 – Tipologie dei messaggi SDP

TIPO DI DESCRITTORE	CARATTERE INDICATORE	PARAMETRI DESCRITTI	OPZIONALE/ OBBLIGATORIO	NOTE
Descrittori e di session	v	Versione del protocollo SDP usato	Obbl.	Deve necessariamente essere la prima riga nel corpo SDP
	o	Creatore della sessione e suo identificativo; indirizzo tipologia della rete; versione del protocollo IP		Obbl.
	s	Nome della sessione (stringa di caratteri)	Obbl.	Non necessario in sessioni VoIP utente-utente. In questo caso è possibile mettere uno spazio bianco dopo "=" in quanto trattasi di un parametro obbligatorio
	i	Informazioni aggiuntive sulla sessione	Opz.	Non necessario se si usa il protocollo SIP
	u	Fornisce un indirizzo URI (Uniform Resource Identifier) per accedere a una risorsa	Opz.	Parametro raramente utilizzato in sessioni VoIP SIP
	e	Indirizzo e-mail	Opz.	
	p	Numero telefonico in formato numerico	Opz.	Inutile in sessioni VoIP SIP
	c	Tipo di connessione; tipo di rete, indirizzo a livello di layer di rete	Opz.	
	b	Banda in Kb/s necessaria per stabilire la connessione	Opz.	
	z	Aggiustamenti dell'orologio dovuti a differenze di fuso orario o ora legale	Opz.	
	k	Dati per la chiave di criptazione	Opz.	
	a	Ulteriori attributi di sessione	Opz.	
Descrittori di tempo	t	Tempo d'inizio (start) e di fine (stop) della sessione	Obbl.	Nel caso di sessioni voce SIP il tempo di fine non è generalmente specificato. In questo caso si usa scrivere "t = 0" che significa durata infinita della conversazione
	r	Per sessioni a programmazione ripetitiva specifica il tempo d'intervallo fra l'una e l'altra	Opz.	Parametro raramente utilizzato in sessioni VoIP SIP
Descrittori dei media utilizzati	m	Tipo di media (audio, video, ecc.); protocollo utilizzato; porta di application multiplexer utilizzata; tipo di payload	Obbl.	Trattasi di un parametro importante di trasporto in ogni sessione SIP

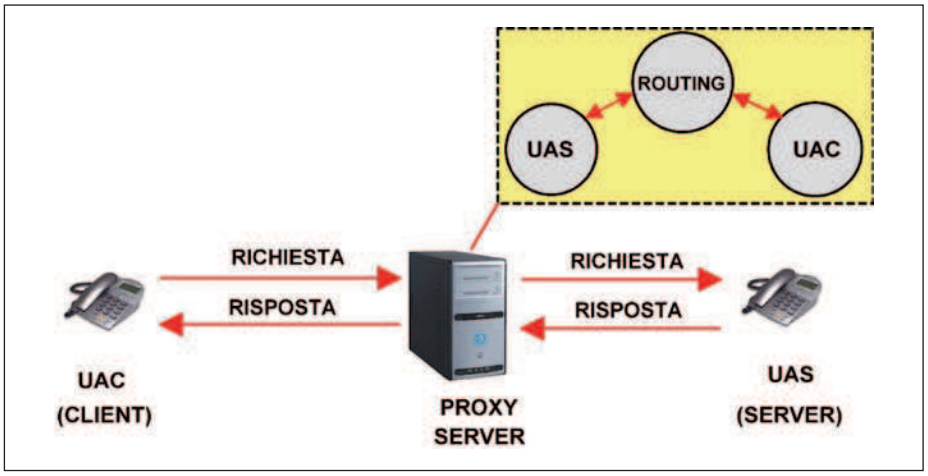


Figura 7: funzioni del proxy server in un'infrastruttura SIP.

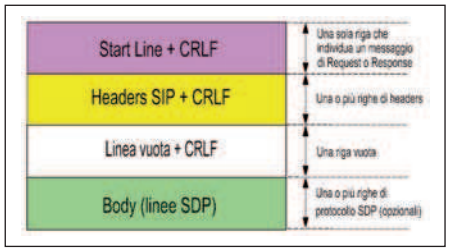


Figura 8: struttura di un messaggio generico SIP.

REDIRECT SERVER

Questo elemento fornisce all'UAC (ovvero il terminale chiamante) tutte le informazioni sui nodi di rete da attraversare e sui percorsi necessari per raggiungere l'UAS (il terminale chiamato). Infatti non sempre l'UAS si trova nell'area coperta dal Proxy Server e, inoltre, il protocollo SIP prevede che lo stesso indirizzo possa essere usato con un endpoint diverso. Questa caratteristica è spesso definita come "nomadicità". Una volta che viene individuato l'UAS il Redirect Server esce di scena e non viene più coinvolto nelle fasi successive della sessione.

REGISTRAR SERVER

Questa entità nella maggioranza dei casi è rappresentata da un apposito programma che risiede nel Proxy o nel Redirect Server. Il compito di un Registrar Server è elaborare i messaggi di registrazione che gli UAC inviano per segnalare la propria posizione all'interno della rete. Se un UAC cambia zona, sfruttando la nomadicità permessa dal protocollo SIP, il Registrar Server ne deve prendere atto e registrare il suo nuovo indirizzo IP in un'apposita lista definita Location Server. Da questo momento in poi l'indirizzo del chiamato sarà associato al suo attuale indirizzo IP. Anche il Location Server è costituito generalmente da un'apposita funzione di un programma in esecuzione. Per comprendere meglio questo concetto occorre sottolineare che in SIP vi è una suddivisione fra unità logiche e unità fisiche. Le prime, costituite dai componenti che abbiamo visto (Proxy, Registrar e Redirect Server), possono anche essere integrate in una sola unità fisica. In questo caso, un solo PC, scelto tenendo conto delle sue prestazioni, potrà svolgere più ruoli ricorrendo a più programmi residenti.

FORMA E SINTASSI DEL PROTOCOLLO SIP

A questo punto possiamo analizzare nei dettagli una sessione voce organizzata sul protocollo SIP. Ricordiamo i punti salienti di questo protocollo:

- la transizione fra UAC e UAS si basa su un modello richiesta-risposta (a ogni richiesta del client deve necessariamente corrispondere una risposta del server);
- i messaggi scambiati sono in testo (molto simile al protocollo HTTP);
- la tipologia del messaggio di richiesta è molto simile al messaggio di risposta, eccetto che per la prima linea di testo;
- ogni richiesta e risposta, oltre all'oggetto della comunicazione, può contenere un message body generalmente costituito da caratteri ASCII o testo in HTML. Scopo dei body è fornire un'ulteriore descrizione, peraltro facoltativa, della sessione in atto. Il body è generalmente costituito da testo, espresso secondo il protocollo SDP.

Queste caratteristiche, che almeno a prima vista dovrebbero consentire una buona interpretazione del protocollo SIP, implicano alcune ristrettezze formali. Vediamo quali:

- il testo dei messaggi è "case sensitive". Ovvero la parola "Pietro" è diversa dalla parola "pietro";
- non possiamo aggiungere spazi bianchi casuali nella prima riga;
- a volte è necessario aggiungere linee vuote (carattere CRLF) se la sintassi lo prevede;
- in alcune circostanze è necessario l'uso di caratteri di interpunzione (generalmente "," oppure ";") nella sintassi.

Abbiamo precedentemente accennato a due protocolli che nella nostra breve trattazione non erano mai stati incontrati: il SAP e l'SDP. Cominciamo con il dire che il SAP (*Session Announcement Protocol*) ha il compito di trasportare l'SDP, ovvero, se preferite, l'SDP è il *payload* del SAP. Da notare che quest'ultimo protocollo, che viene trasmesso in forma binaria, ha bisogno di un "suo" Proxy che, per l'appunto, è definito Proxy SAP. Naturalmente la funzione di un Proxy SAP può essere assolta anche da un Proxy con peculiarità diverse a patto che, nel suo interno, vi sia un apposito software dedicato a questa funzione. Il SAP nasce dall'esigenza di disporre di un protocollo a bassa occupazione di banda, generalmente intorno ai 4 kb/s, destinato al trasporto di annunci di sessioni SIP. Va da sé che se la sessione SIP è già in essere, gli annunci possono essere inviati all'interno di un messaggio SIP e quindi la funzione principale di questo protocollo viene meno. Gli annunci SAP sono ricevuti sempre dalla porta 9875 dell'*Application Multiplexer* e vengono a loro volta trasportati in un *payload* UDP. Vediamo adesso il protocollo SDP. E' questi un protocollo che, diversamente da altri, si esprime non in binario ma con una sintassi di caratteri di tipo testo. La funzione dell'SDP è quella di fornire tutte le informazioni relative ai parametri necessari per stabilire una chiamata voce. Infatti, il protocollo SIP non ha alcuna funzione nativa che permette di far conoscere ad esempio il CODEC usato, la porta dell'*application multiplexer* interessata, la versione del protocollo di rete utilizzato e via dicendo. Solo dopo che UAC e UAS hanno "mediato" le lo-

Problemi di SICUREZZA? Ecco la nostra soluzione!

Vasta scelta di DVR (*Video Registratori Digitali*) per soddisfare qualsiasi esigenza di videosorveglianza.
Tutti i nostri modelli consentono la registrazione in locale ed il controllo remoto tramite Internet.

Set completo per videosorveglianza con monitor LCD da 19"

Si installa in pochi minuti! Possibilità di connessione ad un router ADSL per controllo remoto tramite Internet. Più di tre mesi di registrazione continua sull'hard disk interno.

Il sistema comprende:

- 1 pz DVR 4 canali 100 IPS con telecomando (cod. DVR4L4)
- 1 pz Hard Disk 320 GB SATA
- 1 pz Monitor LCD 19'' WIDE
- 2 pz Telecamera con IR (cod. CP393)
- 2 pz Alimentatore per telecamera
- 2 pz Cavo video con plug d'alimentazione 10 metri (cod. VP002-10)

HDD 320GB SATA

Velocità di trasferimento (7200 rpm) e riduzione degli errori grazie agli 8MB di buffer. Formato 3,5 pollici; rumorosità 32 dB. Ideale per essere utilizzato con i nostri DVR dotati d'interfaccia SATA.

SATAHD320GB

69,00 SERIAL ATA

DVR 4 canali H264 con Ethernet, porta USB e uscita VGA/Videocomposito

SERIAL ATA USB

€ 295,00 IVA INCLUSA

NEW

Videoregistratore digitale real-time con formato di compressione H.264, quattro canali con frame rate di 100 IPS, interfaccia Ethernet (10-100 Base-T), uscita video VGA e videocomposito. Dispone di porta USB, per il trasferimento delle sequenze video su chiavetta, e di ingresso/uscita audio. Funzione Motion Detection, possibilità di registrare, riprodurre e visualizzare contemporaneamente le immagini. Visione da remoto tramite connessione Internet con possibilità di effettuare registrazioni e back-up. Supporta 1 hard disk SATA con capacità massima di 1TB (non compreso).

DVR4L4

Principali caratteristiche dei nostri DVR

Codice		Canali	Interfaccia per Back-up	HDD	Compressione	IPS	Uscita video	IN/OUT audio	Telecom. IR	IN/OUT allarme	Prezzo
FR381		4	Ethernet USB (pendrive)	SATA (interno)	H264	100	V. comp.	4/2	•	4/1	€ 350,00
FR371		4	DVD-RW Ethernet USB (pendrive)	IDE/ATA (interno)	MPEG4 (CIF) / MPEG4 (FRAME)	100	V. comp.	2/1	•	4/1	€ 596,00
FR329AS		4	Ethernet	SATA (interno)	MPEG4 (CIF) / MPEG4 (FRAME)	100	V. comp.	1/1		4/1	€ 264,00
FR375		4	Ethernet USB (pendrive)	SATA (interno)	H264	100	VGA V. comp.	4/1	•	4/1	€ 385,00
FR374		4	Ethernet USB (collegamento PC-Backup)	SATA (interno)	MPEG4 (CIF) / MPEG4 (FRAME)	50	V. comp.	1/1	•	4/1	€ 220,00
FR367AS		4	Ethernet USB (pendrive)	SATA (interno)	MPEG4 (CIF) / MPEG4 (FRAME)	100	V. comp.	1/1	•	4/1	€ 330,00
FR367ASV		4	Ethernet USB (pendrive)	SATA (interno)	MPEG4 (CIF) / MPEG4 (FRAME)	100	VGA V. comp.	1/1	•	4/1	€ 360,00
FR395		8	Ethernet USB (pendrive)	SATA (interno)	H264	200	VGA V. comp.	4/2	•	8/1	€ 840,00
FR382		8	Ethernet USB (pendrive)	SATA (interno)	H264	100	V. comp.	1/1	•	8/1	€ 650,00
FR335		16	CD-RW Ethernet USB (pendrive) RAID	IDE/ATA (interno)	MPEG4 (CIF) / MPEG4 (FRAME)	400	V. comp.	4/2	•	16/1	€ 1.320,00
FR396		16	Ethernet USB (pendrive)	SATA (interno)	H264	400	VGA V. comp.	4/2	•	16/1	€ 1.060,00

NOTA: I videoregistratori digitali vengono forniti senza Hard Disk. Tutti i DVR sono alimentati a bassa tensione mediante specifico adattatore da rete (incluso).
V. comp. = uscita segnale videocomposito (con connettore BNC) per monitor tradizionali di tipo analogico.

CODICE MIP 2753573

FUTURA ELETTRONICA idee in elettronica
Via Adige, 11 • 21013 Gallarate (VA) • Tel. 0331/799775 • Fax. 0331/792287

Maggiori informazioni su questi prodotti e su tutte le altre apparecchiature sono disponibili sul sito *www.futurashop.it* tramite il quale è anche possibile effettuare acquisti on-line.

Tutti i prezzi sono da intendersi IVA inclusa.

Tabella 3 – Protocollo SIP: metodi di richiesta (Request Line)

ACRONIMO	DESCRIZIONE DEL METODO	NORMATIVA DI RIFERIMENTO
INVITE	Indica a un utente (oppure a un servizio) che è stato invitato a una sessione SIP e, avvalendosi del protocollo SDP, descrive la tipologia della sessione stessa	RFC 3261
ACK	Viene inviato dall'UAC che ha già trasmesso un INVITE per confermare che la sessione è stabilita	
BYE	Termina una sessione e può essere inviato sia dal chiamante che dal chiamato	
CANCEL	Elimina qualunque richiesta ancora pendente (anche non risolta)	
OPTIONS	Usato da un UA che chiede di conoscere le prestazioni e/o caratteristiche di un Server o di un altro UA	
REGISTER	Usato da un UA generico per annotare la propria presenza su un Register Server	
PRACK	Riconoscimento (ACK) provvisorio. Usato se vengono interfacciate reti PSTN ove sono possibili problemi di segnalazione	RFC 3262
SUBSCRIBE	Permette di registrarsi presso una lista di UA che, in seguito, saranno interessati a eventi/messaggi inviati loro tramite il metodo NOTIFY	RFC 3265
UNSUBSCRIBE	Permette di cancellarsi da una lista di UA, vedi metodo SUBSCRIBE	
NOTIFY	Notifica un evento/messaggio a un UA generico. È usato in stretta relazione con il metodo SUBSCRIBE	
PUBLISH	Annuncia un evento (sessione SIP) a un Server	RFC 3903
INFO	Invia informazioni inerenti la sessione in corso che tuttavia non possono modificare lo stato della sessione stessa. Usato, ad esempio, per le informazioni di tariffazione e per inviare toni DTMF	RFC 2976
REFER	Necessario per il trasferimento di chiamata. Supponiamo che A e B abbiano una sessione in corso e A decida di trasferire la chiamata a C. Per ottenere ciò A dovrà inviare un REFER a B	RFC 3515
MESSAGE	Invia un messaggio generico usando il protocollo SIP. Può sostituire la classica e-mail	RFC 3428
UPDATE	Modifica lo stato di una sessione senza cambiare il dialogo in corso fra UAC e UAS	RFC 3311

ro esigenze ricorrendo al protocollo SDP sarà possibile il transito dei pacchetti in RTP, questi ultimi contenenti i campioni audio del parlato. E' importante sapere che le informazioni fornite da SDP riguardano essenzialmente tre categorie:

- sessione, ovvero tutte le informazioni relative ai contenuti della sessione (se voce, video, oppure conferenza, ecc.);
- tempistica, vale a dire i tempi d'inizio conversazione, di durata, ecc. Questi dati sono a volte necessari per la tariffazione;
- trasmissione, da intendersi come informazioni riguardanti i flussi da ricevere e trasmettere (CODEC, numero campioni al secondo, porte utilizzate, ecc.).

La tipologia di un "body" (insieme di istruzioni) in SDP è molto semplice e vale la pena descriverla. Essa si basa sempre su una serie di istruzioni che occupano una riga ognuna. La sintassi è:

"tipo" = "valore"

in cui "tipo" è un semplice carattere ("a", "v", ecc.) che individua univocamente il parametro che stiamo descrivendo. L'utilizzo di un singolo carattere alfanumerico è una scelta obbligata dal fatto che SDP è un protocollo testuale cosicché, in quanto tale, occupa sicuramente una banda maggiore rispetto ad altri protocolli che si esprimono in forma binaria. Con la dicitura "valore" indichiamo una stringa alfanumerica che definisce il valore numerico o gli attributi assunti dal parametro in oggetto. Il carattere "=", sempre presente, indica la separazione fra le due entità. Può essere interessante notare che le informazioni SDP possono essere inviate in *multicast* (un solo utente trasmette verso più utenti riceventi) nel caso la sessione SIP riguardi, ad esempio, una conferenza. In SDP non tutti i parametri sono obbligatori, per quelli opzionali la scelta della loro presenza dipende dalla nostra intenzione di rendere la sessione VoIP più efficiente e "robusta". Le righe SDP devono comunque essere immesse secondo un ordine prestabilito, indipendentemente dalla presenza o meno di campi facoltativi. In **tabella 2** troviamo l'elenco commentato di tutte le tipologie dei messaggi SDP.

A questo possiamo entrare nel dettaglio della struttura di un messaggio SIP, tenendo conto, ancora una volta, della suddivisione concettuale fra messaggi di richiesta (*Request*) e di responso (*Response*). Il corpo di un messaggio SIP è mostrato in **figura 8**. Osservando la figura possiamo notare come le linee dei primi tre campi debbano essere necessariamente terminate con il carattere CRLF, ovvero il tasto RETURN nei sistemi WINDOWS e MS-DOS. Inoltre, la differenza fra messaggio di *Request* e di *Response* è individuabile soltanto dalla Start Line (prima riga). Infatti il resto del messaggio, body compreso, può essere identico nella forma, seppure differente nel contenuto delle singole linee testuali. La convenzione vuole che nel caso di messaggio di *Request* la prima riga venga chiamata *Request Line.* Nel caso di messaggio di risposta essa assume invece il nome di *Status Line*. Il protocollo SIP definisce quindici tipologie di messaggi di richiesta (alcuni preferiscono chiamarli "metodi"), tuttavia, nella pratica corrente, ne vengono usati generalmente sei. In **tabella 3** possiamo trovare tutti i metodi di richiesta con relativo commento esplicativo. Si noti, osservando la **tabella 3**, che i metodi di uso più comune sono definiti nella raccomandazione RFC 3261.

Veniamo ora alle varie tipologie di risposta cominciando con il definire lo schema fisso della *Status Line* (prima riga in un messaggio di risposta). Una *Status Line* deve essere sempre composta dai seguenti elementi (il segno "+" non fa parte della sintassi del messaggio):

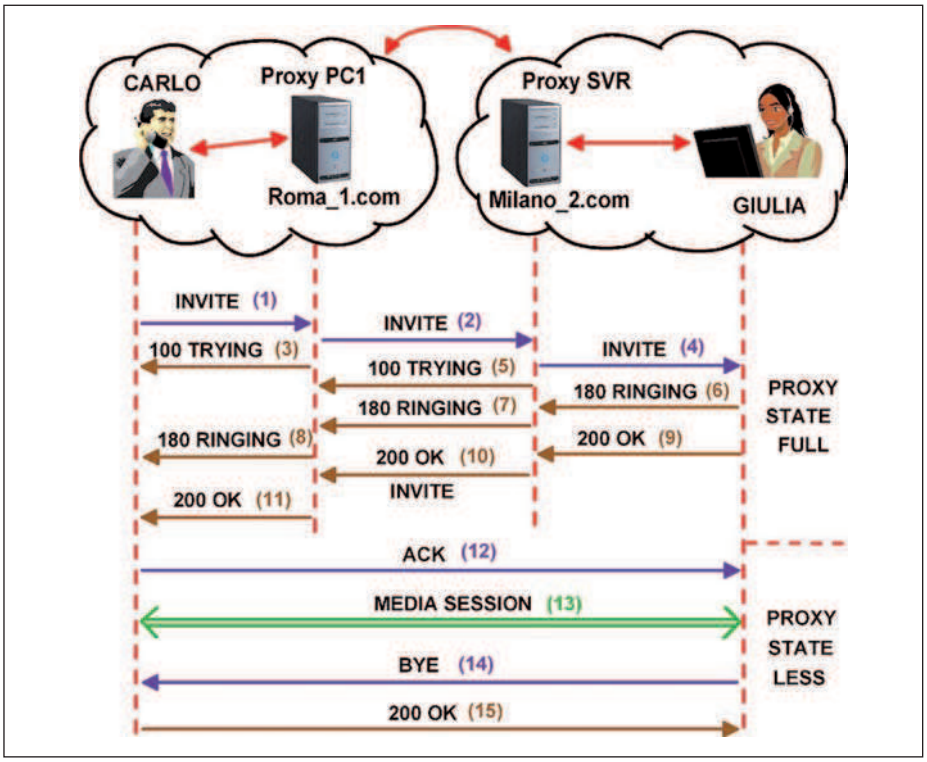


Figura 9: esempio di chiamata SIP fra due utenti.

Versione SIP + Codice numerico + Frase esplicativa + CRLF

Analizziamo ora i singoli campi. "Versione SIP" ovviamente si riferisce alla versione di protocollo usata, che generalmente è la 2.0. Il campo "Codice Numerico" è formato da un numero a 3 cifre compreso, per la versione SIP 2.0, fra 100 e 699. Di questo numero la prima cifra indica la classe della risposta mentre le altre due cifre individuano l'esatto messaggio o parametro. Essendo sei le classi di risposta (da 1xx a 6xx), ci si potrebbe aspettare che ognuna possa contenere 99 messaggi, come logica vorrebbe. In realtà molte combinazioni sono prive di significato poiché i progettisti dello standard, pensando a future implementazioni, si sono lasciati una piccola riserva. La "Frase esplicativa" (chiamata anche *Reason Phrase*) è una breve descrizione alfabetica del significato della risposta. A ogni codice numerico è associata una precisa *Reason Phrase* e quindi non dobbiamo inventarci noi le parole da aggiungere. In **tabella 4** sono riassunte tutte le classi di risposta con qualche esempio significativo.
Per ogni messaggio di risposta ricevuto, l'UAC deve a sua volta rispondere inviando un ACK. Esiste una sola eccezione: i *Responses* della serie 1xx che essendo riferiti a situazioni provvisorie non prevedono necessariamente una risposta di conferma. Una volta definita la Start Line possiamo passare alla descrizione dei cosiddetti *header* (vedi sempre **figura 8**). Il compito delle linee di *header* è quello di fornire informazioni aggiuntive, ma decisamente necessarie, per lo svolgimento di una sessione SIP (indirizzi IP dell' UAC e UAS, identificativo numerico della sessione, ecc.). Per complicare un po' le cose c'è da dire che alcune righe di *header* possono essere composte da più sottocampi (in questo caso un carattere ";" fa da separatore) e alcuni *header* possono essere aggiunti solo a determinate tipologie di messaggio SIP. In **tabella 5** possiamo trovare le quattro classi di *header* in ragione della loro compatibilità con i messaggi di *Request* e di *Response*. Nella stessa tabella, per ogni classe, è stata aggiunta una breve spiegazione della funzionalità dei campi di competenza e l'elenco, non commentato, di questi ultimi. La descrizione, anche superficiale, di ogni singola riga di *header* esula dai scopi di questa trattazione.
La sintassi di una riga di *header* è molto semplice:

Field Name : Field Value

dove "Field Name" identifica il parametro e "Field Value" ne fornisce il valore numerico (o alfabetico). Se una riga di *header* non è riconosciuta, ad esempio per un errore di sintassi, essa sarà ignorata ma non è detto che a questo debba seguire necessariamente un messaggio di errore da parte del ricevente. Il carattere ":" fa ovviamente da separatore e deve essere sempre presente. Le righe di *header* possono essere immesse senza un preciso ordine ma si consiglia, per ragioni di velocità di processo, di mettere per prime quelle che forniscono dati di *routing* (percorso dei pacchetti dati).

ESEMPIO DI SESSIONE VOCE SIP

Una volta definiti nel dettaglio i singoli costituenti di un messaggio SIP, sia esso un *Request* o un *Response*, possiamo approfondire lo svolgimento di una semplice sessione voce fra due utenti. In **figura 9** l'utente Carlo prova a chiamare l'utente Giulia usando il protocollo SIP. Per rendere la simulazione più veritiera immagineremo che il flusso dei messaggi, prima di avviare lo stream voce, debba attraversare due Proxy. La presenza di questi ultimi simula una condizione molto frequente nell'accesso a Internet. Generalmente, infatti, per disporre delle risorse di rete siamo tenuti a sottoscrivere una forma di abbonamento con un gestore di servizi (Provider). Questi ci fornirà un cosiddetto dominio di appartenenza il cui nome sarà una parte, ad esempio, del nostro indirizzo e-mail e dell'identificativo SIP per l'utente. Far parte di un dominio significa, da un punto di vista hardware, essere registrati presso uno o più server. Anche questi ultimi, a loro volta, hanno un nome identificativo (da non confondere con il loro indirizzo!) così come riportato nel nostro esempio (PC1 e SVR).
Per meglio comprendere lo svolgimento della chiamata, in **figura 9** abbiamo riportato la Start Line di ogni messaggio SIP prodotto dai due utenti. Per semplicità di trattazione l'*header* e il body del messaggio non vengono descritti. Essendo Carlo il chiamante, o UAC che dir si voglia, avrà come Start Line del suo primo messaggio una *Request Line.* Commentiamo adesso le varie sequenze della chiamata (indicate con i numeri fra parentesi).

Tabella 4 – Protocollo SIP: tipologie di risposta (Status Line)

CODICE DI RISPOSTA	CLASSE DEL MESSAGGIO	ESEMPI DI RISPOSTE COMMENTATE
1xx Risposta provvisoria (deve seguire necessariamente una risposta definitiva)	181 call forwarded 180 ringing	Chiamata inviata Il telefono del chiamato sta squillando
2xx Richiesta eseguita con successo	200 OK	Riposta positiva del chiamato
3xx Deviazione della chiamata a un altro indirizzo (specificato nella risposta)	302 user mover temporarily	Il chiamato risponde presso un altro indirizzo
4xx Richiesta non eseguita	401 user not authorized	L'utente non è autorizzato per questo servizio
	486 busy	L'utente è impegnato in altra conversazione
5xx Guasto del Server	500 server internal error	Errore nel funzionamento del server
6xx Problema di connessione generico (per esempio l'utente chiamato non esiste)	604 does not anywhere	Utente o servizio impossibile da localizzare

(1) Carlo emette una richiesta di INVITE indirizzata a Giulia, la richiesta è ricevuta dal Proxy Roma_1.com. Nel messaggio di Carlo sono contenute tutte le informazioni (CODEC, versioni dei protocolli, ecc.) che permettono all'utente Giulia di stabilire una connessione voce su IP con protocollo SIP.
(2) Il Proxy Roma_1.com a sua volta emette una richiesta di INVITE, contenente anch'esso l'indirizzo di Giulia, che viene ricevuto dal Proxy Milano_2.com.
(3) Il Proxy Roma_1.com emette un altro messaggio, questa volta di 100 TRYING, verso Carlo. Lo scopo è quello di rispondere a quest'ultimo indicandogli che la chiamata sta procedendo.
(4) Il Proxy Milano_2.com emette una richiesta di INVITE che viene direttamente ricevuto dal PC di Giulia, dotato di *softphone*.
(5) Il Proxy Milano_2.com emette un messaggio di risposta tipo 100 TRYING verso il PROXY Roma_1.com. Anche in questo caso lo scopo della risposta è comunicare al server Roma_1.com che la chiamata è attiva e procede.
(6) Il *softphone* di Giulia comincia a squillare e questo viene comunicato al server Milano_2.com con una risposta di 180 RINGING. La risposta contiene l'indirizzo di Carlo poiché è destinata a lui.
(7) Il proxy Milano_2.com ripete una risposta di 180 RINGING verso il server Roma_1.com. Anche questa risposta contiene l'indirizzo di Carlo.
(8)Il server Roma_1.com trasmette una risposta di 180 RINGING direttamente al telefono SIP di Carlo. Quest'ultimo comincerà a sentire il tono di chiamata.
(9) Il *softphone* di Giulia, dopo aver analizzato le "proposte" (CODEC, versione protocolli, banda utilizzata, ecc.) contenute nella richiesta di INVITE fatta da Carlo, ha deciso che è possibile stabilire uno stream voce con quest'ultimo. Risponde quindi con un 200 OK diretto al Server Milano_2.com. In realtà le fasi (6) e (9) sono contemporanee. Infatti, non verrebbe emessa una risposta di 180 RINGING da Giulia se non vi fosse accordo con le "proposte" di Carlo.
(10) La riposta 200 OK è ricevuta dal Server Milano_2.com che provvede a "girarla" al Server Roma_1.com.
(11) Il Server Roma_1.com trasmette la risposta 200 OK direttamente al telefono SIP di Carlo. Da questa fase in poi la funzionalità dei Server sarà esclusivamente passiva, diremo che sono passati da un funzionamento State Full a quello definito come State Less (vedi paragrafo SIP Proxy Server). Potremmo anche dire che Carlo e Giulia, da questa fase in poi, sono in collegamento punto-punto (o, se preferite, pier to pier).
(12) Il telefono di Carlo invia una richiesta ACK che stavolta giungerà direttamente a Giulia. Il *request* ACK indica all'UAS che lo stream voce può iniziare.
(13) Inizia la sessione audio durante la quale è comunque consentito a entrambi gli utenti di ricevere e trasmettere messaggi SIP.
(14) Giulia chiude la conversazione e il suo *softphone* invia una *request* di tipo BYE a Carlo.
(15) Il telefono SIP di Carlo prende atto della *request* e risponde con un 200 OK per avvisare Giulia che non vi sono problemi e la conversazione è chiusa.
L'esempio riportato è forzatamente semplice, seppure frequente, ed esclude situazioni pure possibili come ad esempio quando l'utente chiamato non risponde o le caratteristiche hardware di uno dei due utenti non rendono possibile la conversazione. Essendo il SIP un protocollo che si esprime in forma testuale, è possibile riportare un esempio completo di messaggi di *request* e di *response* e, sia pur nei limiti della nostra trattazione, commentarli. In **figura 10** e **figura 11** sono mostrate due tipologie di messaggi sicuramente fra le più frequenti nel protocollo SIP. Per continuità espositiva entrambe si ricollegano all'esempio di rete VoIP mostrato in **figura 9**. Analizziamo adesso le righe che costituiscono il messaggio di *request* contenuto in **figura 10**:

- INVITE sip: Giulia@Milano_2.com SIP/2.0: l'utente Giulia, che è registrata presso il server Milano_2.com, viene invitata a una sessione SIP. La versione del protocollo usata è la 2.0;
- Via: SIP/2.0/UDP PC1.Roma_1.com; *branch* z9hG4bKabrtjs5: viene specificato il protocollo di trasporto utilizzato (in questo caso UDP) e fornito l'indirizzo del Proxy server di Carlo (l'ente a cui sarà diretta la *Response*). Sono inoltre forniti dei dati che possono essere utilizzati nel processo di *routing* dei pacchetti dati (*branch* parameters), la stringa "z9hG4bK" posto subito dopo il parametro "*branch*", è invece necessaria per individuare la versione SIP e viene definita "Magic cookie";
- Max-Forwards: 70: massimo numero di nodi (Gateway, Server, ecc.) che il messaggio di *Request* può attraversare;
- From: Carlo <Carlo@Roma_1.com>; tag=1927573125: nome e indirizzo di colui che ha originato il messaggio di richiesta (nel nostro caso Carlo). Segue un numero casuale (tag) generato per meglio identificare il richiedente (Carlo);
- To: Giulia <sip: Giulia@Milano_2.com>: nome e indirizzo dell'utente a cui è effetti-

```
INVITE sip: Giulia@Milano_2.com SIP/2.0
Via: SIP/2.0/UDP
PC1.Roma_1.com; branch= z9hG4bKabrtjs5
Max-Forwards: 70
From: Carlo <Carlo@Roma_1.com>; tag=1927573125
To: Giulia <sip: Giulia@Milano_2.com>
Call-ID: a75b3c66e55310@PC1.Roma_1.com
CSeq: 1 INVITE
Date: Mon, 9 Mar 2009 19:05:05 GMT
Contact: <sip:Carlo@PC1.Roma_1.com>
Content-Type: application/sdp
Content-Lenght: 142

v= 0
o= Carlo 52655664 2313643637 IN IP4 PC1.Roma_1.com
s= Session SDP
t= 0 0
c= IN IP4 PC1.Roma_1.com
m= audio 3456 RTP/AVP 0 1 3 99
a= rtpmap: 0 PCMU/8000
```

Request Line

Righe di headers

Una riga vuota

Righe di body (SDP)

Figura 10: esempio di Request tipo INVITE.

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP SVR.Milano_2.com; branch= z7hG4bKeruihj8
Via: SIP/2.0/UDP PC1.Roma_1.com; branch= z7hG4bKabrtjs5
From: Carlo <Carlo@Roma_1.com>; tag=1927573125
To: Giulia <sip: Giulia@Milano_2.com>
Call-ID: a75b3c66e55310@PC1.Roma_1.com
CSeq: 1 INVITE
Contact: <sip:Carlo@PC1.Roma_1.com>
Content-Type: application/sdp
Content-Lenght: 142

v= 0
o= Carlo 52655664 001 IN IP4 SVR.Milano_2.com
s= stream audio
t= 0 0
c= IN IP4 SVR.Milano_2.com
m= audio 3456 RTP/AVP 0 1 3 99
a= rtpmap: 0 PCMU/8000
```

Status Line

Righe di headers

Una riga vuota

Righe di body (SDP)

Figura 11: esempio di Response tipo 200 OK.

vamente trasmesso il messaggio di INVITE (nel nostro caso Giulia);

- Call-ID: a75b3c66e55310@PC1.Roma_1.com: numero che identifica in modo univoco la chiamata in corso. Esso è generato da una combinazione ottenuta da un numero casuale ed elementi presenti nella chiamata (indirizzo IP, ecc.);
- CSeq: 1 INVITE: numero sequenziale che identifica la transazione (*Request* e *Response*) in corso. Esso si incrementa per ogni nuova *Request*. Viene inoltre ripetuto il tipo di richiesta (in questo caso un INVITE);
- Date: Mon, 9 Mar 2009 19:05:05 GMT: data e ora della richiesta fornite rispetto al GMT;
- Contact: <sip: Carlo@PC1.Roma_1.com>: indirizzo e dominio di Carlo. Può anche essere espresso sotto forma di indirizzo IP;
- Content-Type: application/sdp: descrive la tipologia delle righe costituenti il body del messaggio di *Request*. Nel nostro esempio il body è espresso in SDP;
- Content-Lenght: 14 tipo di CODEC usato (in questo caso G.711 µlaw con 8000 campionamenti al secondo). 2 = Numero di byte contenuti nel body.

Dopo una riga vuota appare il body del messaggio, in questo esempio è composto da righe testo scritte con protocollo SDP:

- v= 0: versione del protocollo SDP usato;
- o= Carlo 52655664 001 IN IP4 PC1.Ro-

Tabella 5 – Protocollo SPI: classi e tipi di headers

CLASSE	DESCRIZIONE	COMPATIBILITÀ CON MESSAGGI DI REQUEST	COMPATIBILITÀ CON MESSAGGI DI RESPONSE	TIPI DI HEADERS	
General Headers	Specifica gli indirizzi IP del chiamato e del chiamante, fornisce indirizzi per la ridirezione delle chiamate, ecc.	SI	SI	Call-ID CSeq Encryption From Timestamp	Contact Date Expires Record-Route To Via
Request Headers	Contiene campi dedicati alla parametrizzazione delle richieste (urgenza, priorità, ecc.)	SI	NO	Accept Accept-Language Contact Max-Forwards Priority Proxy-Require Require Subject	Accept-Encoding Authorization Hide Organization Proxy-Authorization Route Response-Key User-Agent
Response Headers	Contiene campi dedicati alla parametrizzazione delle risposte e fornisce dati sulla disponibilità dei servizi richiesti	NO	SI	Allow Retry-After Unsopported WWW-Authenticate	Proxy-Authenticate Server Warning
Entity Headers	Descrive in dettaglio il contenuto delle righe che compongono il body (generalmente fornite in protocollo SDP)	SI	SI	Content-Encoding Content-Lenght Content-Type	

ma_1.com: nome dell'utente che ha originato il messaggio SIP, numero identificativo della sessione in corso e dell'attuale messaggio, tipologia della rete, versione del protocollo IP, nome del server Proxy di Carlo e dominio,

- s= stream audio: indica l'oggetto della sessione, in questo caso uno stream audio (voce utenti);
- t= 0 0: tempo d'inizio e di fine della sessione voce. Il secondo zero indica che il tempo di fine non è determinato poiché, in una sessione voce, non è possibile sapere con esattezza il momento della chiusura. Entrambi i tempi sono espressi in secondi NTP (Network Time Protocol);
- c= IN IP4 PC1.Roma_1.com: tipo di connessione (Internet), tipo e versione del protocollo di rete usato e nome e indirizzo IP del Server di Carlo;
- m= audio 3456 RTP/AVP 0 1 3 99: tipologia del media (in questo caso voce, quindi audio), porta dell'*Application Multiplexer* utilizzata, protocollo di trasporto e tipo di *payload*;
- a= rtpmap: 0 PCMU/8000: tipo di CODEC usato (in questo caso G.711 µlaw con 8000 campionamenti al secondo).

Vediamo adesso in dettaglio il messaggio di *response* contenuto in **figura 11**. Cominciamo dalle righe di *header*:

- SIP/200 OK: l'utente Giulia ha accettato l'invito a partecipare alla sessione voce (telefonata) proposta da Carlo. L'invio di questa *Response* coincide con l'atto di alzare il microtelefono;
- Via: SIP/2.0/UDP SVR.Milano_2.com; *branch*= z7hG4bKeruihj8: viene specificato il protocollo di trasporto utilizzato (anche in questo caso UDP) e l'indirizzo del primo server che il messaggio di *Response* incontrerà per arrivare a Carlo;
- Via: SIP/2.0/UDP PC1.Roma_1.com; *branch*= z7hG4bKabrtjs5: questo *header* indica l'indirizzo del secondo e ultimo server che il messaggio incontrerà prima di arrivare a destinazione. Da notare che quando questa *Response* sarà processata dal server SVR.Milano_2.com il primo *header* "Via" sarà cancellato in quanto non più necessario;
- From: Carlo Carlo@Roma_1.com; tag= 1927573125: anche in questo *header* viene fornito il nome e l'indirizzo di colui che ha originato il messaggio di richiesta (nel nostro caso Carlo). Anche il tag è identico a quello di "INVITE";
- To: Giulia <sip: Giulia@Milano_2.com>: è' indicato il medesimo indirizzo presente nello stesso *header* del messaggio di "INVITE". In altri termini viene conservata la direzione logica della richiesta;
- Call-ID: a75b3c66e55310@PC1.Roma_1.com: il "Call-ID deve essere lo stesso del messaggio di "INVITE";
- CSeq: 1 INVITE: l'*header* riporterà lo stesso numero sequenziale del messaggio di "INVITE" in quanto ci troviamo all'interno della stessa transazione;
- Contact: <sip: Carlo@PC1.Roma_1.com>: è indicato sempre l'indirizzo e il dominio di Carlo;
- Content-Type: application/sdp: descrive la tipologia delle righe costituenti il body del messaggio di *Request*. Nel nostro esempio il body è espresso in SDP;
- Content-Lenght: 142: Numero di byte contenuti nel body.

Dopo una riga vuota troviamo il body del messaggio. In questo esempio è composto da righe in protocollo SDP, così come nella *Request*.

- v= 0: versione del protocollo SDP usato;
- o= Carlo 52655664 001 IN IP4 SVR.Milano_2.com: nome dell'utente che ha originato il messaggio SIP, numero identificativo della sessione in corso e dell'attuale messaggio, tipologia della rete, versione del protocollo IP, nome del server Proxy di Giulia e dominio;
- s= stream audio: indica l'oggetto della sessione, in questo caso uno stream audio (voce utenti);
- t= 0 0: tempo d'inizio e di fine della sessione voce, anche in questo caso la durata è indeterminata (parametro "0");
- c= IN IP4 SVR.Milano_2.com: tipo di rete (Internet), tipo e versione del protocollo di rete usato e nome e indirizzo IP (in forma letterale) del Server di Giulia;
- m= audio 3456 RTP/AVP 0 1 3 99: tipologia del media (in questo caso voce quindi audio), porta dell'*Application Multiplexer* utilizzata, protocollo di trasporto e tipo di *payload*;
- a= rtpmap:0 PCMU/8000: tipo di CODEC usato (anche in questo caso G.711 µlaw con 8000 campionamenti al secondo).

CONCLUSIONI

Siamo dunque arrivati alla fine di questa trattazione che, per forza di cose, poteva essere solo introduttiva. Prima di chiudere vorrei dare qualche consiglio a coloro che intendono continuare lo studio dei sistemi VoIP. Immagino la prima domanda che i futuri sistemisti si faranno: fra SIP e H323 quale conviene portare avanti? La risposta è molto semplice. Se non esistono motivi particolari (assistenza su impianti preesistenti, implementazioni del software, ecc.) per cui l'architettura H323 debba essere studiata, converrà puntare sul SIP. Quest'ultimo si sta dimostrando, nelle sue varie versioni, un protocollo versatile, sicuro e di facile implementazione. E queste sono caratteristiche che, alla lunga e a detta di molti, porteranno alla completa obsolescenza il suo "rivale" H323. D'altro canto, i numerosi impianti tuttora funzionanti con questo standard e la quantità di hardware prodotta, faranno sì che, per i prossimi 2-3 anni, l'H323 non scompaia del tutto. Lo studio pratico del SIP, che tra l'altro si presta bene a un approccio didattico, può essere fatto implementando un sistema VoIP sul proprio PC. Esistono appositi gateway in grado di rendere un comune telefono per rete PSTN adatto a funzionare come terminale VoIP con standard SIP. Il costo di questi dispositivi, che hanno connettività a livello di LAN, è tale da renderne l'acquisto assolutamente fattibile. Per non parlare poi dei *softphone*, ormai reperibili sotto forma di modelli somiglianti a un comune portatile. Il software che simula le altre entità SIP è poi rintracciabile facilmente su Internet, spesso con le limitazioni di una versione gratuita di prova a tempo limitato. A facilitare ulteriormente lo studio "autodidatta" ci sono poi dei provider che offrono abbonamenti gratuiti per telefonia SIP. L'analisi vera e propria del protocollo va invece demandata ai vari analizzatori di traffico, o sniffer, che sono dei programmi in grado di mostrare sul monitor del nostro PC il contenuto dei pacchetti dati. Ne esistono molti per cui ne citeremo solo alcuni: il sempreverde Wireshark che ha sostituito ormai il vecchio (ma ancora efficiente) Ethereal, il Network Monitor della Distinct reperibile in versione gratuita a tempo e infine Lanwatch anch'esso reperibile in versione demo. Buon lavoro! ❑

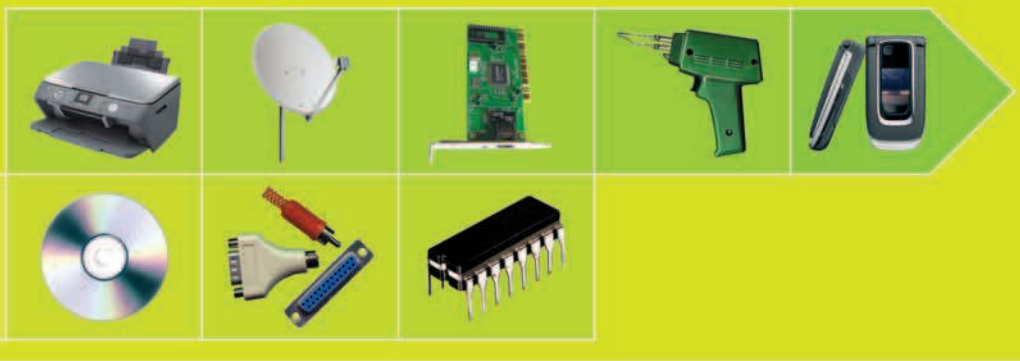
CODICE MIP 2757039

Con il Patrocinio di
Ministero dello Sviluppo Economico
Comune di Modena

Expo Elettronica

MOSTRA MERCATO

■ informatica ■ elettronica ■ materiali di consumo
■ dvd e games ■ telefonia ■ hobbistica



MODENA FIERE

Viale Virgilio, 70/90

23 - 24 gennaio 2010 • ore 9/18

Collezionsa®

MOSTRA MERCATO SCAMBIO

■ collezionismo ■ modellismo ■ hobby ■ curiosità

DOPO LE FESTE, GRANDI AFFARI!

VALE COME RIDOTTO

fotografa questo codice e scarica il ticket ridotto direttamente sul tuo smartphone

SNAP IT

NEWS ON LINE WWW.EXPOELETTRONICA.IT e iscriviti a NEWSLETTER. 10 giorni prima di un evento riceverai una e-mail di promemoria

Organizzazione e informazioni BLU Nautilus www.expoelettronica.it tel. +39 0541 439573

CODICE MIP 2778647

rif. Fare Elettronica

Lezioni di VHDL *(parte ottava)*

SILICORE SU SPARTAN

Nella scorsa puntata ci siamo soffermati sul processore SLC1657, un RISC a 8 bit. Il processore, realizzato interamente in VHDL e diffuso da Silicore con la solita licenza GPL, risulta particolarmente indicato per applicazioni di controllo

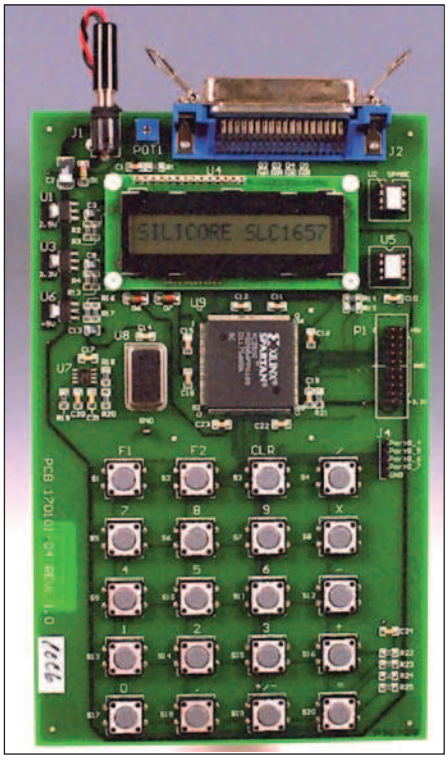


Figura 1: EVB Spartan 2.

L'obiettivo di un progettista di una qualsiasi applicazione VHDL è quello di eseguire il proprio programma al pari di un modulo software. In realtà, in questo contesto è improprio utilizzare il termine programma; infatti, ci stiamo riferendo a un progetto scritto in VHDL e perciò squisitamente hardware e non di un modulo software. A questo scopo per "eseguire" un linguaggio VHDL è necessario inserire l'applicazione in un FPGA o in un ASIC, vale a dire una volta terminato il processo di sintesi è necessario fare il cosiddetto *place&route* della nostra applicazione. Questo è il filo conduttore dell'ultima parte del corso. In effetti, in questa puntata vogliamo mettere in evidenza le attività che sono necessarie per integrare il processore SLC1657 in un FPGA. In altre parole, si vuole proporre la realizzazione di un SoC (System-on-Chip) partendo dal core SLC1657 che chiameremo XSP2EVAL, Xilinx SPartan 2 EVALuation system. SoC è un termine utilizzato per indicare dispositivi elettronici nei quali viene integrato in un solo chip un intero sistema, vale a dire che oltre al classico processore sono presenti diversi controller, come porte di I/O o sistemi video.

Il processore SLC1657, come abbiamo già messo in evidenza nella scorsa puntata, è compatibile con il PIC16C57 della Microchip Technology, e in questo modo possiamo utilizzare i vari ambienti di sviluppo già in uso con Microchip. Come ambiente di lavoro abbiamo scelto l'evaluation kit della Spartan II; l'evaluation board è mostrata in **figura 1**.

Il SoC che andremo a realizzare ha la necessità di utilizzare diverse entity VHDL. Alcune dovremo definirle perché risultano indispensabili alla nostra applicazione, mentre altre sono già presenti nel core del processore di Silicore, come ad esempio toplogic. La prima entity è chiamata XSP2EVAL, Xilinx Spartan 2 Evaluation. La sua rappresentazione gerarchica è mostrata in **figura 2**. Seguono poi TOPLOGIC, REGISRAM (register RAM), INSTRROM, SEMRMINT e i driver di I/O per Xilinx Spartan II quali IBUF, OBUF e IOBUF.

Il nostro lavoro è diviso in varie fasi. Nella prima dovremo mettere a punto il processore SLC1657 e simulare l'entità TOPLOGIC. Questo lavoro permetterà all'utilizzatore di familiarizzare con l'ambiente di simulazione, con il core SLC1657 e con tutte le componenti del progetto. È bene ricordare che questa particolare attività risulta indipendente dal target utilizzato e si presta anche per altri componenti di Altera o Xilinx. Nell'attività di simulazione si è utilizzato il Protel PeakVHDL simulation, ma ad ogni modo possiamo pensare di utilizzare un qualsiasi strumento di simulazione VHDL.

XSP2EVAL

Che cosa occorre fare per simulare il core SLC1657? Per prima cosa è necessario creare una cartella di lavoro, per esempio 'TL TEST'. In seguito, una volta aperto PeakVHDL, definiamo un nuovo progetto denominato, appunto, 'TL TEST'. Il progetto così creato deve riferire la cartella omonima. In seguito dobbiamo aggiungere tutti i moduli dell'entità TOPLOGIC nel progetto, facendo attenzione a rispettare la gerarchia delle entity. È opportuno rimarcare il rispetto della gerarchia delle entità. Infatti, per svolgere il lavoro di simulazione con PeakVHDL dobbiamo essere sicuri di rispettare il grado gerarchico delle entità e a questo scopo utilizzeremo il test bench di TOPLOGIC, TSTBENCH.VHD, che troveremo nella cartella TOPLOGIC del codice sorgente del core. Un estratto del testbench è mostrato nel **listato 1**. Inoltre è necessario utilizzare i vari test vector corrispondenti che troveremo nella stessa cartella e individuati con l'estensione txt. Tutti questi file dovranno essere posizionati nella cartella 'TL TEST'. La **figura 3** mostra i file contenuti nel progetto. Una volta copiati tutti i file nel progetto possiamo senz'al-

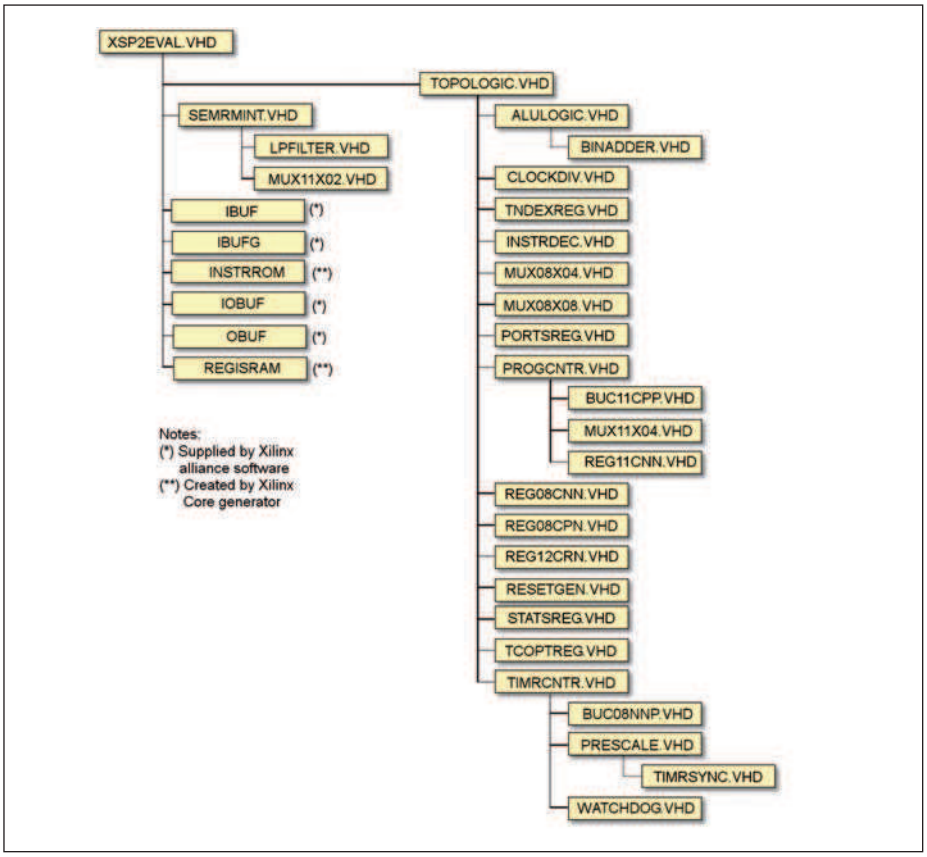


Figura 2: ESP2EVAL.

tro iniziare il processo di simulazione che deve concludersi senza nessun errore. Il **listato 1** pone in evidenza un estratto del file di testbench e le relazioni con i test vector. A questo proposito l'istruzione:

```
file_open( tvfile, "VECTI-
NIT.TXT", read_mode );
```

apre il file 'vectinit.txt' dove sono presenti alcuni parametri di simulazione, mentre con:

```
READ_VECTORS: loop
```

si inizia il ciclo 'READ_VECTORS' utilizzato per impostare il nostro testbench, con le relative variabili, con i valori del vettore di test. Il **listato 2** mostra un possibile formato di un file di questo tipo, in questo caso si impostano i valori del reset, del registro index e delle porte di I/O.
Così, il file pone in evidenza che il registro index è inizializzato a 0x4, vale a dire che il registro è localizzato alla locazione 0x4 della memoria interna, mentre i registri di I/O occupano le locazioni da 0x5 a 0x7, rispettivamente PORT0, PORT1 e PORT2.
Una volta terminata la simulazione del nostro core, occorre definire l'entity VHDL REGISRAM (register RAM), una memoria sincrona a 128x8 bit. L'entità VHDL è necessaria perché occorre definire la zona di memoria utilizzata dai registri interni del processore. La **tabella 1** pone in evidenza la struttura della memoria interna utilizzata dal core. Nel panorama commerciale esistono diversi sistemi per definire una memoria di questo tipo, ma il procedimento più semplice passa attraverso il sistema automatico di generazione fornito dalla maggior parte di costruttori di sistemi di sviluppo su FPGA.
Xilinx a questo proposito fornisce il core generator in grado di creare e definire le caratteristiche delle memorie. Per prima cosa occorre creare l'entity REGISRAM. Dopo aver creato una cartella, che chiameremo 'REGISARM', apriamo Xilinx core generator e impostiamo il tool al fine di creare una RAM sincrona nella cartella appena creata.
I parametri utilizzati per impostare, in maniera appropriata, questa sequenza sono messi in evidenza nel **listato 3**. In seguito possiamo generare REGISRAM in automatico e verifichiamo che la nella cartella omonima è presente un file chiamato 'regisRAM.edn': in questo modo otteniamo la conferma che la procedura si è conclusa senza problemi con il file EDIF per la memoria.
Terminata la fase di generazione della memoria RAM interna dobbiamo fare la stessa cosa per ottenere l'entity INSTRROM (instruction ROM). Questa particolare entity è utilizzata come embedded oppure emulation ROM a scopo di sviluppo software. In INSTRROM dovranno essere caricati i vari moduli software, in formato binario, ottenuti dai tool di sviluppo di casa Microchip. Così, il registro program counter del processore SLC1657 deve contenere l'indirizzo dell'Instruction ROM.
I programmi saranno caricati attraverso la porta parallela del PC utilizzando l'entity LPFILTER del SoC XSP2EVAL, parte integrante del progetto VHDL. La memoria di programma è composta di 2.048 word a 12 bit. La memoria è organizzata in quattro banchi da 512 word. Il processore SLC1657 utilizza la tecnica a banchi perché la sua architettura si basa su 12 bit e, per questa ragione, non è possibile gestire indirizzamenti di memoria più lunghi di 9 bit. La selezione del banco di lavoro è permessa attraverso i bit IB1-IB0 del registro di stato (status, si veda la **figura 4**).
Nella versione Xilinx Spartan II, l'instruction ROM è formata con blocchi sincroni di memoria RAM. L'INSTRROM, anche in questo caso, è definita attraverso lo Xilinx Core Generator tool. Per creare la nostra entity INSTRROM è necessario, per prima cosa, creare una cartella chiamata 'INSTRROM'. In seguito, una volta aperto il tool Xilinx Core Generator, occorre impostare i parametri per creare una RAM sincrona.
I parametri utilizzati sono mostrati nel **listato 4**. Una volta configurato l'ambiente è possibile procedere per generare l'entity INSTRROM in maniera automatica e, anche in questo caso, verifichiamo, al termine della procedura, che nella cartella omonima è presente il file 'instrrom.edn': il file EDIF per l'instrucion ROM.

LISTATO 2 – Vectinit.txt

```
_____________________________________
— Reset using the 'RESET' pin.
_____________________________________

:  100 000 00 00 00 00 7FF 0 UU 00 0 UU UU UU FF FF FF 000  RESET
:  000 000 00 00 00 00 7FF 0 UU 00 0 UU UU UU FF FF FF 000  RESET
:  000 000 00 00 00 00 7FF 0 UU 00 0 UU UU UU FF FF FF 000  NOP

_____________________________________
— Initialize the INDEX register.
_____________________________________

:  000 C00 00 00 00 00 000 0 UU 00 0 UU UU UU FF FF FF 000  MOVI 0x00
:  000 024 00 00 00 00 001 0 S4 00 0 UU UU UU FF FF FF 000  MOVA 0x04

_____________________________________
— Initizlize the I/O ports.
_____________________________________

:  000 C00 00 00 00 00 002 0 00 00 0 UU UU UU FF FF FF 000  MOVI 0x00
:  000 025 00 00 00 00 003 0 05 00 0 UU UU UU FF  FF FF 000  MOVA 0x05
:  000 026 00 00 00 00 004 0 06 00 0 00  UU UU FF  FF FF 100  MOVA 0x06
:  000 027 00 00 00 00 005 0 07 00 0 00  00  UU  FF FF FF 010  MOVA 0x07
:  000 005 00 00 00 00 006 0 05 00 0 00  00  00   FF FF FF 001  MOVP 0x05
:  000 006 00 00 00 00 007 0 06 00 0 00  00  00   00 FF FF 000  MOVP 0x06
:  000 007 00 00 00 00 008 0 07 00 0 00  00  00   00 00 FF 000  MOVP 0x07
:  000 000 00 00 00 00 009 0 00 00 0 00  00  00   00 00 00 000  NOP

_____________________________________
— Branch to the very top of memory, on the last bank.  Clear the
— INDEX and STATUS registers.  This places the register and
— instruction banks to zero.  End the file with the program counter
— at 0x7FF so that the program counter rolls over so that the first
— instruction of the next file will be at address 0x000.
_____________________________________

:  000 C60 00 00 00 00 00A 0 00 60 0 00 00 00 00 00 00 000  MOVI 0x60
:  000 023 00 00 00 00 00B 0 03 60 0 00 00 00 00 00 00 000  MOVA 0x03
:  000 BFC 00 00 00 00 00C 0 1C FC 0 00 00 00 00 00 00 000  BRA  0x7FC
:  000 000 00 00 00 00 7FC 0 1C FC 0 00 00 00 00 00 00 000  NOP
:  000 C00 00 00 00 00 7FD 0 00 00 0 00 00 00 00 00 00 000  MOVI 0X00
:  000 023 00 00 00 00 7FE 0 03 00 0 00 00 00 00 00 00 000  MOVA 0x03
:  000 024 00 00 00 00 7FF 0 04 00 0 00 00 00 00 00 00 000  MOVA 0x04
```

LISTATO 3 – Regisram

```
Device type: Spartan 2
File format: VHDL
Tool type: Other (Protel)
Netlist bus format: B(I)
Memory type: distributed
Component name: regisram
Depth: 128
Data width: 8
Memory type: single port RAM
MUX construction: LUT based
Input options: non-registered
Layout: create RPM
```

LISTATO 4 – instrrom

```
Device type: Spartan 2
File format: VHDL
Tool type: Other (Protel)
Netlist bus format: B(I)
Memory type: single port block
memory
Component name: instrrom
Depth: 2048
Data width: 12
Port configuration: read and
write
Global init value: 0 (leave the
initialization file box
unchecked).
```

Sintesi

Il passo successivo del nostro lavoro è la sintesi. Si è già scritto che il nostro progetto è gerarchico e per questa ragione il file in VHDL 'XSP2EVAL' è l'entità che racchiude tutte le definizioni del nostro progetto. Dal **listato 5** vediamo che TOPLOGIC è un modulo del nostro SoC. Per svolgere il nostro lavoro di sintesi ricorriamo al Protel PeakVHDL synthesis.

Per prima cosa creiamo la cartella 'XSP2EVAL', apriamo PeakVHDL e definiamo un nuovo progetto seguendo a questo scopo tutte le indicazioni mostrate a video. Inseriamo tutti i moduli definiti dalla entity XSP2EVAL nel progetto, rispettando le relazioni gerarchiche. Le entità richiamate in TOPLOGIC possono essere trovate nella cartella 'VHDL_source', mentre quelle in

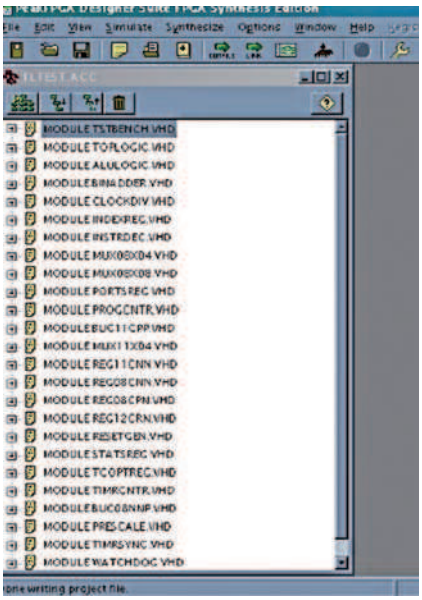


Figura 3: progetto VHDL.

XSP2EVAL (come ad esempio SEMRMINT.VHD) sono posizionate nella cartella 'Xilinx'. I nostri due file che abbiamo creato nei punti precedenti, vale a dire regisram.edn e instrrom.edn, devono essere sposati nella cartella XSP2EVAL. Si selezioni l'opzione 'Spartan 2 Series (EDIF)' presente nel tool PeakVHDL e successivamente si attiva il processo di sintesi. È opportuno, al termine del processo di sintesi, controllare il file di log per verificare l'esito del processo. Al termine del processo di sintesi verifichiamo la presenza del file 'XSP2EVAL.edn': è un EDIF file creato da PeakVHDL.

Place & Route

A questo punto siamo giunti nella fase di *place&route* del nostro design nel chip Xilinx Spartan II. Il file EDIF 'XSP2EVAL', ottenuto nella precedente fase, contiene la maggior parte della circuiteria di controllo utile al nostro scopo, insieme al modulo software Xilinx Alliance Series. Una volta lanciato l'Alliance design manager, dobbiamo creare un nuovo progetto selezionando 'XSP2EVAL.EDN' come file di input (lo stesso file ottenuto nel passo precedente). Nella sezione 'Part Selector' occorre impostare le seguenti opzioni:

- Family: SPARTAN2;
- Device: XC2S50;
- Package: PQ208;
- Speed Grade: -5.

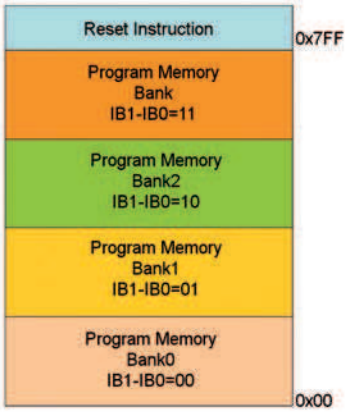


Figura 4: memoria a banchi.

PER *approfondire...*

i listati 1 e 5 sono disponibili su www.farelettronica.com

Successivamente, nella sezione 'Constraints File' si selezioni l'opzione 'Custom' definendo un file di configurazione del nostro progetto. Questo file deve contenere alcune informazioni, per esempio le specifiche dei tempi e l'allocazione dei pin. Il file di solito ha l'estensione UCF, ad esempio 'XSP2EVAL.UCF'.

CONCLUSIONE

Siamo così giunti alla fine di questo corso sul linguaggio VHDL. Si è cercato di esporre il linguaggio attraverso esempi applicativi in maniera semplice e chiara, ma, come per tutti i corsi, l'argomento non può esaurirsi; infatti, per comprendere appieno un linguaggio occorre che ogni appassionato realizzi una propria applicazione. A questo scopo diversi costruttori propongono strumenti di lavoro liberi da costi che ne permettono l'uso in ambiente didattico e a scopo personale. In quest'ultima puntata abbiamo messo in evidenza un'applicazione utile e interessante al tempo stesso, si sono mostrate le varie fasi indispensabili per integrare il microcontrollore SLC1657 in un FPGA al fine di ottenere un SoC (System-on-Chip) chiamato XSP2EVAL, Xilinx SPartan II EVALuation system. Dal sito della rivista è possibile prelevare il materiale di lavoro del core SLC1657: dal codice VHDL a tutta la documentazione di riferimento. ❑

CODICE MIP 2757079

CODICE MIP 2771344

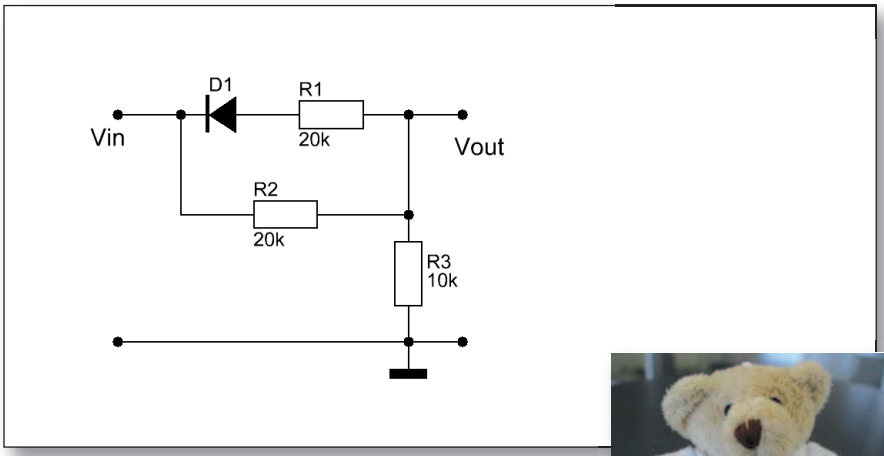
elettro quiz 294/295

rispondi & vinci

*Le risposte ai quiz "Base" e "Avanzato" vanno inviate esclusivamente compilando il modulo su www.farelettronica.com/eq specificando la parola chiave **"2N2222"**.*

Le risposte ed i vincitori (previa autorizzazione) sono pubblicati alla pagina www.farelettronica.com/eq a partire dal 15 del mese successivo alla pubblicazione sulla rivista.

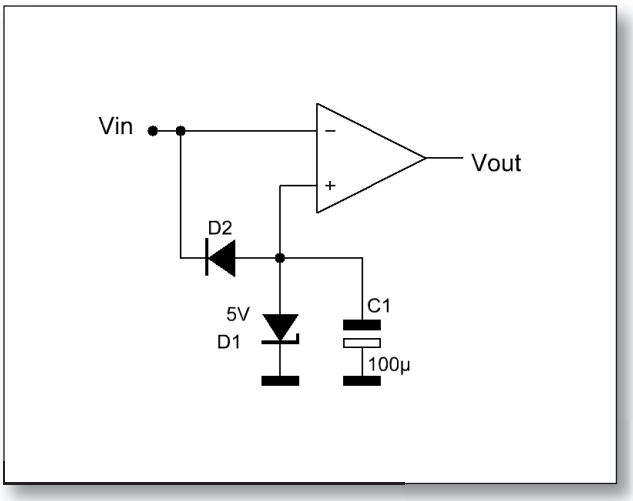
A tutti i partecipanti verrà assegnato un buono sconto del 10% (validità 3 mesi dalla data di assegnazione) utilizzabile per un prossimo acquisto su www.ieshop.it



base

Dato il circuito in figura determinare il valore della tensione di uscita quando all'ingresso vengono applicati rispettivamente 12V e -12V. Si consideri ideale il comportamento del diodo.

Se rispondi correttamente potrai vincere il simpatico portachiavi di Fare Elettronica.



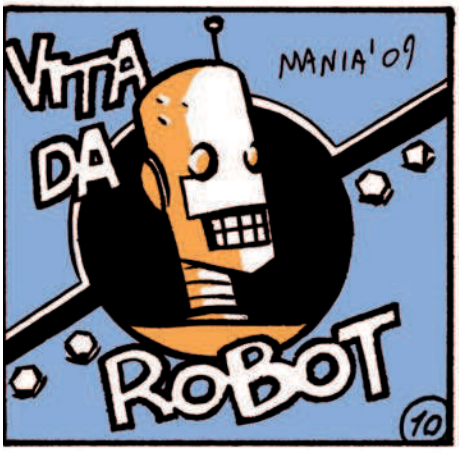
avanzato

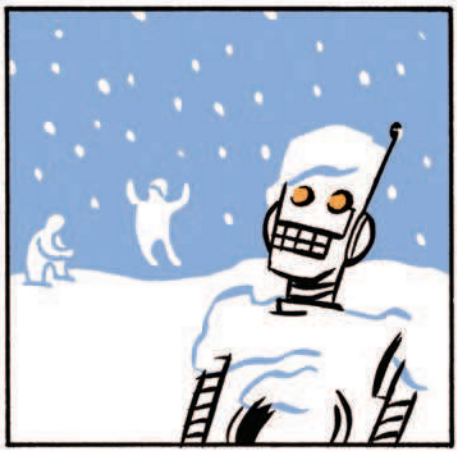
Il circuito mostrato in figura è costituito utilizzando un amplificatore operazionale, un diodo e uno zener da considerarsi ideali. Quale sarà la tensione di uscita se all'ingresso viene applicata una sinusoide di ampiezza 10V e frequenza 1KHz?

Per i più bravi in palio la felpa pile di Fare Elettronica.

luditronica

Le vignette ludiche di Fare Elettronica





TRASMETTITORE DSB-SC per i 20 metri

Se siete in procinto di prendere la patente di radioamatore e volete fare un po' di pratica di laboratorio prima di affrontare gli esami, ecco un piccolo progetto che vi dà la possibilità di testare su campo quanto avete studiato sui libri

Non c'è niente di più efficace per fissare i concetti fondamentali della radio-tecnica che passare un po' di tempo in laboratorio montando e testando qualche semplice apparecchiatura. Costruire un oscillatore quarzato, vedere in funzione un modulatore bilanciato, assemblare un trasformatore a larga banda, sono tutte cose che un radioamatore patentato dovrebbe saper fare, se non altro perché comunque fanno parte delle nozioni di base richieste per il superamento degli esami e il conseguimento della patente.
La realizzazione del circuito descritto di seguito ha proprio lo scopo di vedere in pratica il funzionamento di alcuni degli elementi fondamentali che compongono un apparecchio radio; in particolare vedremo come realizzare un piccolo trasmettitore a modulazione di ampiezza con portante soppressa e doppia banda laterale, utilizzando in tutto solo tre transistor e una manciata di componenti passivi.

LA MODULAZIONE DI AMPIEZZA

Storicamente la modulazione di ampiezza è stata la prima forma di modulazione per la trasmissione di messaggi di fonia come voce, suoni o musiche. I difetti principali di questo tipo di modulazione sono l'elevata sensibilità ai disturbi elettrici e la bassa efficienza. Dal punto di vista formale un segnale radio modulato in ampiezza da un segnale audio sinusoidale può essere rappresentato con la seguente espressione:

$$v_{(t)} = V_p>[1 + m_a\cos(\omega_m t)]\cos(\omega_p t) \qquad (1)$$

dove $\omega_m = 2\pi fm$ è la pulsazione angolare del segnale sinusoidale modulante; $\omega_p = 2\pi fp$ è la pulsazione angolare del segnale radio portante, con $\omega_p >> \omega_m$; ma = Ka(Vm/Vp) è l'indice di modulazione e deve essere minore di 1 affinché l'inviluppo del segnale modulato abbia lo stesso andamento dell'informazione da trasmettere.
Per ma>1 il segnale v(t) si dice in sovramodulazione. In tal caso si introducono notevoli distorsioni nell'inviluppo del segnale modulato che non consentono, in ricezione, una ricostruzione fedele dell'informazione. Nella normale radiodiffusione ma = 40%. Vm e Vp sono rispettivamente l'ampiezza del segnale modulante e della portante radio, mentre Ka è una costante di proporzionalità legata al modulatore utilizzato. Applicando qualche formula di trigonometria, la (1) può essere riscritta:

$$v_{(t)} = V_p*\cos(\omega_p t) + (m_a V_p/2)\mathrm{COS}(\omega_p-\omega_m)t+ + (m_a V_p/2)\mathrm{COS}(\omega_p+\omega_m)t \qquad (2)$$

In pratica v(t) risulta essere pari alla composizione di tre segnali in cui il primo non è altro che la portante $[V_p*\cos(\omega_p t)]$ e gli altri due sono segnali detti "bande laterali" disposti simmetricamente rispetto alla portante sull'asse delle frequenze (vedi **figura 1**). Da notare che:

- l'informazione del segnale modulante è presente solo nelle bande laterali;
- il rendimento di modulazione definito come rapporto tra la potenza del segnale di una delle due bande laterali e la potenza complessiva è pari al 16,7% (con $m_a = 1$). In pratica la maggior parte della potenza espressa non contribuisce al trasporto dell'informazione;

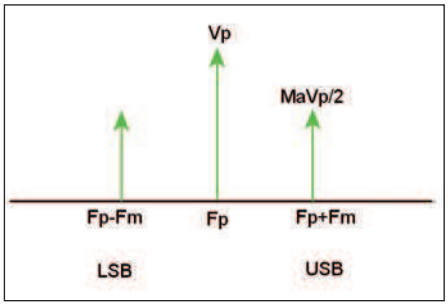


Figura 1: lo spettro di un segnale modulato in ampiezza.

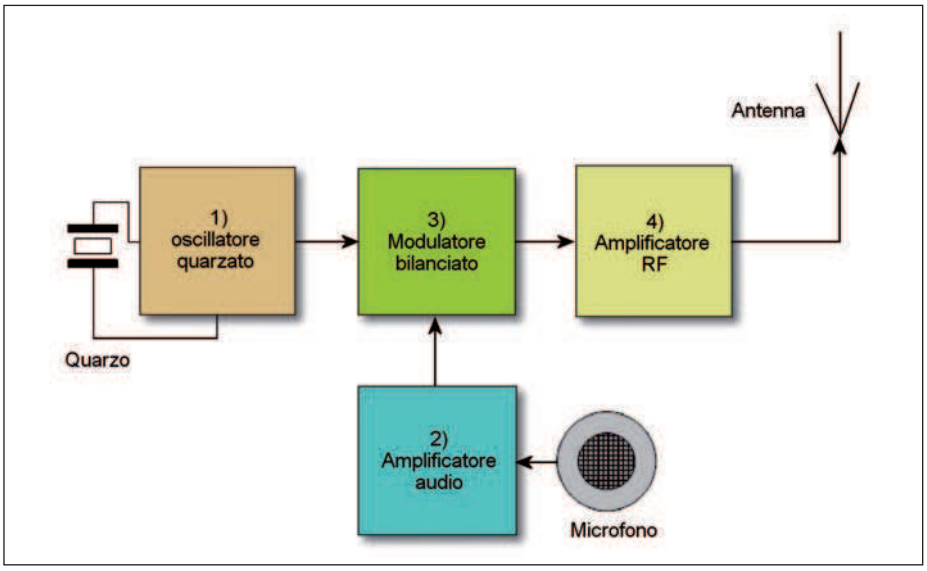


Figura 2: schema di principio del trasmettitore.

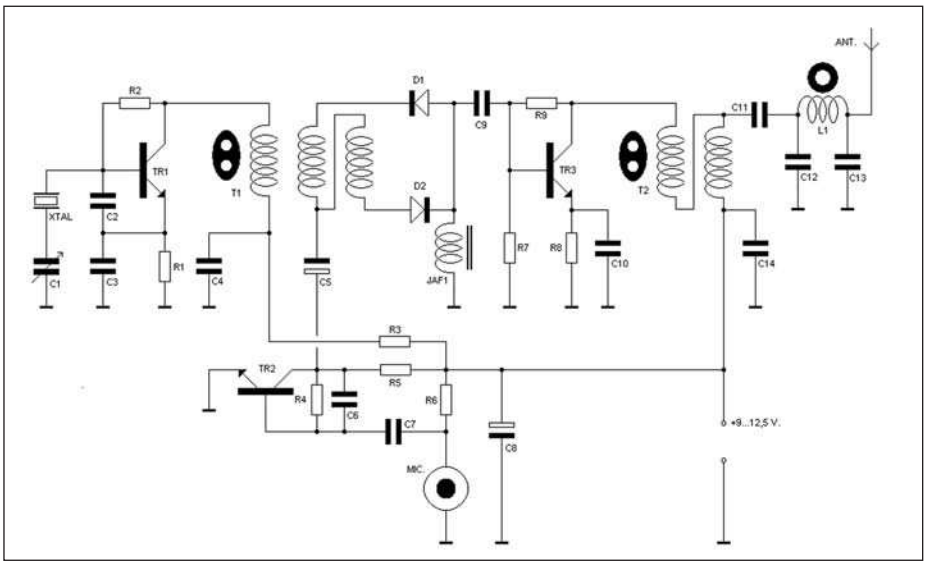


Figura 3: schema elettrico complessivo del trasmettitore.

• la banda occupata dal segnale radio modulato è pari al doppio della banda occupata dal segnale modulante. Se ad esempio prendiamo in considerazione un segnale audio complesso con un'occupazione di banda 20 Hz-20 kHz la larghezza del canale radio per trasmettere tale segnale dovrebbe essere di circa 40 kHz.

Per rendere la modulazione di ampiezza più efficiente, sono state introdotte altre due tecniche che mirano ad aumentare il rendimento di modulazione e diminuire l'occupazione di banda.

La prima tecnica è la DSB-SC (Double Side Band – Suppressed Carrier) che prevede la generazione di un segnale modulato in ampiezza, privo della portante, che come abbiamo visto non contribuisce al trasporto dell'informazione. In questo caso la (2) può essere scritta:

$$(m_aV_p/2)COS(\omega_p-\omega_m)t + (m_aV_p/2)COS(\omega_p+\omega_m)t$$

Il rendimento di modulazione passa dal 16,7% al 50%, ma l'occupazione della banda rimane sempre pari al doppio di quella occupata dal segnale modulante. La seconda tecnica SSB-SC (Single Side Band – Suppressed Carrier) ha lo scopo di ottimizzare anche la banda occupata. Considerando, infatti, che l'informazione da trasmettere è duplicata nelle due bande laterali, è sufficiente trasmetterne una, così che, con la modulazione SSB, oltre a raggiungere un rendimento di modulazione pari al 100%, la banda occupata dal canale radio è esattamente pari a quella del segnale modulante. In questo caso la (3) diventa:

LSB-SC $(m_aV_p/2)COS(\omega_p-\omega_m)t$ (Lower Side Band)

USB-SC $(m_aV_p/2)COS(\omega_p+\omega_m)t$ (Upper Side Band).

Non a caso la SSB è la modalità di trasmissione in fonia maggiormente utilizzata in ambito radioamatoriale.

SCHEMA ELETTRICO

In **figura 2** è visibile lo schema di principio del nostro trasmettitore e in **figura 3** lo schema elettrico complessivo, come si può vedere i blocchi funzionali sono sostanzialmente quattro, ovvero:

• l'oscillatore quarzato, che ha il compito di generare la portante radio;
• l'amplificatore audio, che incrementa il segnale captato dal microfono;
• il modulatore bilanciato, che determina la modulazione di ampiezza con soppressione della portante;
• l'amplificatore finale RF.

Il cuore del nostro trasmettitore è il blocco numero 3, ovvero il modulatore bilanciato. La versione proposta è molto semplice, in pratica fa uso di un trasformatore a larga banda opportunamente avvolto e di due diodi al silicio. Ai capi dell'impedenza JAF1, in assenza di modulazione, il segnale della portante generato dall'oscillatore quarzato è sostanzialmente nullo; non appena viene applicato il segnale audio, questo va ad agire sulla conduzione dei due diodi, generando in uscita a tutti gli effetti un segnale DSB-SC ovvero privo della componente relativa alla portante. Ma veniamo agli altri blocchi. L'oscillatore quarzato è una classica configurazione Colpitts, il quarzo de-

LISTA COMPONENTI

R1	100 ohm ¼ W
C1	7-20 pF, compensatore
C10	2,2 nF ceram.
D1	1N4148
R2	100 kohm ¼ W
C2	220 pF ceram.
C11	100 nF pol.
D2	1N4148
R3	15 ohm ¼ W
C3	220 pF ceram.
C12	120 pF ceram
TR1	2N706
R4	220 kohm ¼ W
C4	100 nF pol.
C13	220 pF ceram.
TR2	BC550
R5	1,8 kohm ¼ W
C5	4.7 uF 25 V elettr.
C14	100 nF pol.
TR3	2N3725
R6	1,5 kohm ¼ W
C6	220 pF ceram.
JAF1	100 uH
XTAL	14.318 MHz
R7	1 kohm ¼ W
C7	220 nF pol.
L1	Vedi testo
MIC.	Microfono preamplific.
R8	15 ohm ¼ W
C8	10 uF 25 V elettr.
T1	Vedi testo
R9	2,2 kohm ¼ W
C9	100 nF pol.
T2	Vedi testo

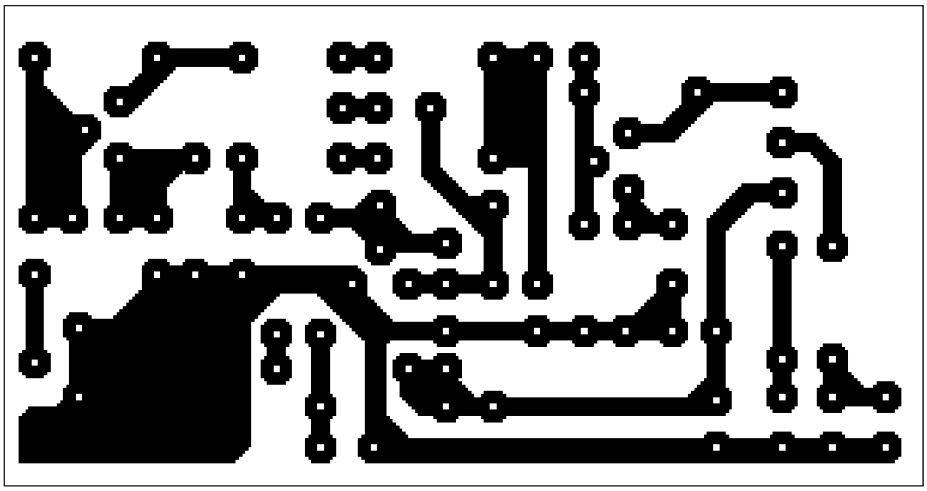


Figura 4: circuito stampato per l'assemblaggio.

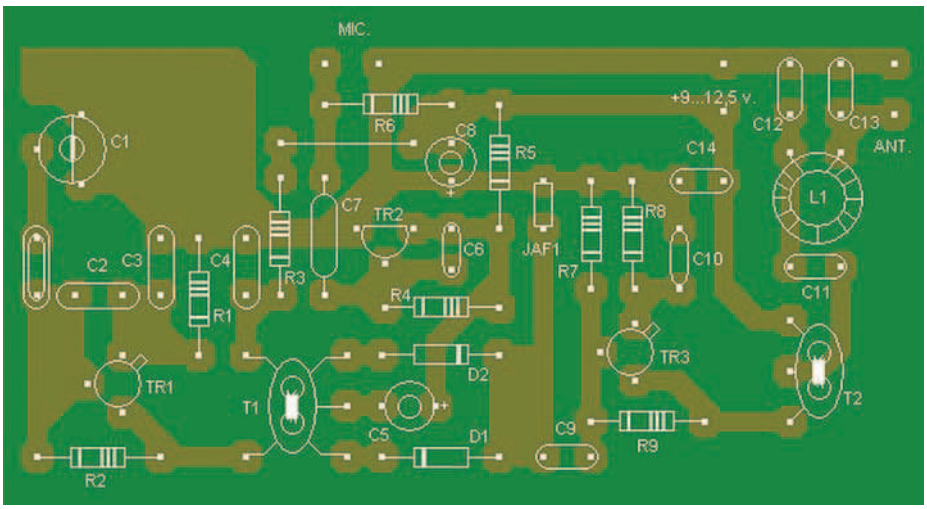


Figura 5: indicazioni del piano di montaggio.

termina la frequenza di oscillazione. Nel prototipo è stato utilizzato un quarzo per computer da 14.318 MHz, valore che cade nella banda dei 20 metri assegnata ai radioamatori. Inoltre tramite il compensatore C1 sarà possibile regolare di qualche kHz la frequenza generata. L'amplificatore audio, composto dal transistor TR2 e componenti limitrofi, tramite il condensatore C5, porta il segnale prodotto dal microfono al modulatore bilanciato. Il segnale DSB-SC presente all'uscita del modulatore bilanciato viene amplificato dallo stadio successivo composto da TR3 configurato come amplificatore in classe A. Sul collettore di TR3 troviamo un secondo trasformatore a larga banda che assieme al filtro passa basso composto da L1, C12 e C13 migliora l'adattamento di impedenza con il circuito d'antenna da 50 ohm.

MONTAGGIO E TARATURA

Il circuito non risulta particolarmente critico, e in realtà non ci sono tarature da effettuare, ma la realizzazione dei trasformatori a larga banda e della bobina L1 deve essere fatta con cura.
Se non si è già sufficientemente esperti di montaggi in alta frequenza, si consiglia l'utilizzo di un apposito circuito stampato per l'assemblaggio, come quello proposto in **figura 4**, seguendo le indicazioni del piano di montaggio di **figura 5**. I transistor possono essere sostituiti con degli equivalenti; la sostituzione di TR3 può determinare una minore potenza in uscita che, con i componenti indicati e un'alimentazione di 12,5 volt, può arrivare a circa 250 mW picco-picco.Qui di seguito viene spiegato come procedere con l'avvolgimento degli induttori.

Trasformatore a larga banda T1

Prendete tre spezzoni di filo di rame smaltato da 0,3 mm, lunghi circa 50 cm e attorcigliateli in maniera simmetrica; è importante che l'attorcigliamento sia uniforme, per fare questa operazione si consiglia l'uso di un semplice trapanino avvitatore a bassa velocità. Utilizzate il filo attorcigliato per realizzare il trasformatore, avvolgendo in una ferrite binoculare tipo Amidon "BN 43-1502", 10 spire; il numero di spire non è critico, 1 o 2 spire in più o in meno non pregiudicano il buon funzionamento del trasformatore. Effettuato l'avvolgimento, allargate i 3 fili componenti i due terminali e con l'aiuto di un tester, contrassegnate gli avvolgimenti come visibile in **figura 6**, unite assieme quindi i terminali B0 e C1; la presa così realizzata sarà quella da col-

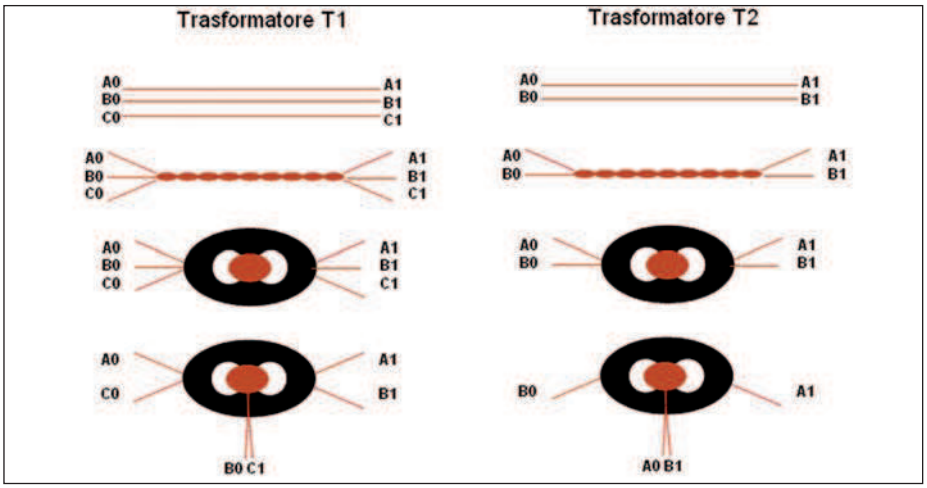


Figura 6: la realizzazione dei due trasformatori.

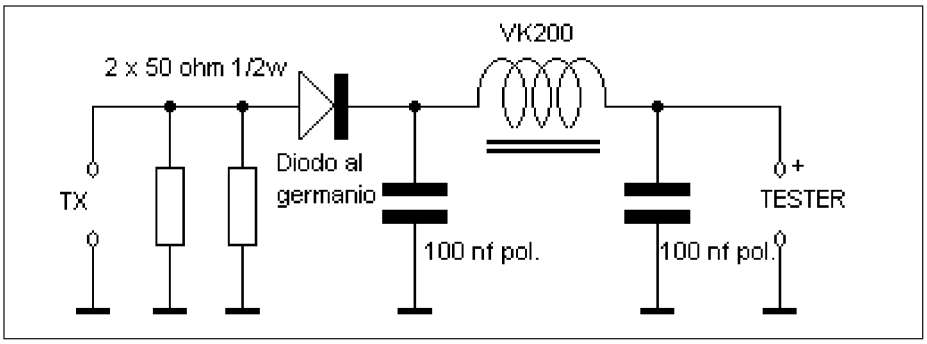


Figura 7: indicazioni per realizzare una piccola sonda di carico.

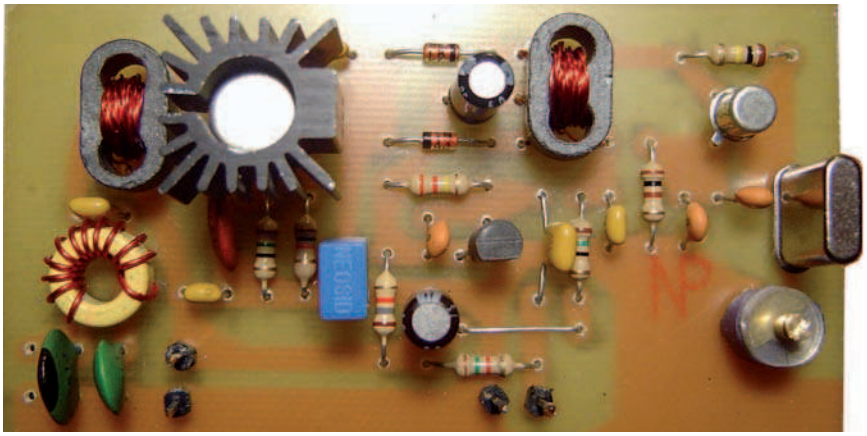
legare al condensatore C5, mentre gli estremi C0 e B1 andranno collegati ai due diodi al silicio.

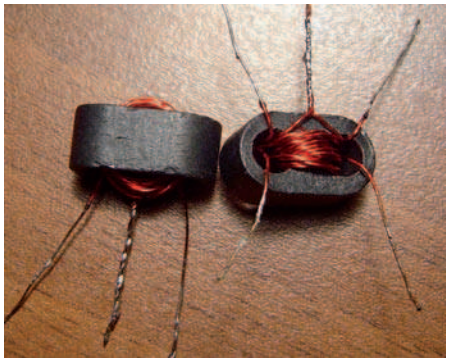
Trasformatore a larga banda T2

Prendete due spezzoni di filo di rame smaltato da 0,3 mm, lunghi circa 50 cm e procedete come per T1 all'attorcigliamento. Utilizzate il filo così preparato per avvolgere sempre in una ferrite binoculare tipo Amidon "BN 43-1502" ,13 spire, anche qui il numero non è critico. Allargate i fili componenti i due estremi del trasformatore e sempre con l'ausilio di un tester marcate i 4 terminali unite A0 e B1, ottenendo così la presa che sarà collegata a C11.

Bobina L1

L1 va realizzata avvolgendo 12 spire di filo di rame smaltato da 0,5 mm in un toroide Amidon di colore giallo T-37-6.





Procedimento

Procedete assemblando sul circuito stampato le varie parti e ricordate di collocare su TR3 un piccolo dissipatore. Come microfono potete utilizzare una piccola capsula preamplificata o uno di quei microfoni che si collegano al PC. Collegate al posto dell'antenna una piccola sonda di carico, se non ne possedete una la potete facilmente realizzare seguendo le indicazioni dello schema di **figura 7**. Collegate alla sonda un tester posizionato su 20 volt fondo scala, date quindi tensione al circuito. Se tutto è stato fatto bene, in assenza di modulazione sul tester dovreste leggere un valore di tensione molto bassa, dell'ordine di 0,3-0,5 volt, mentre parlando a voce alta sul microfono dovreste raggiungere picchi di tensione pari a 5-6 volt. La piccola presenza di portante è dovuta al fatto che i due diodi al silicio del modulatore bilanciato, non sono perfettamente uguali, bisognerebbe sceglierne una coppia con stessa resistenza inversa e diretta per ottenere una tensione quasi nulla in assenza di modulazione o passare a un modulatore bilanciato più complesso. Non provate ad ascoltare il segnale trasmesso su una comune radiolina in grado di captare le onde corte, perché almeno che non abbia integrato un BFO o un comunque un demodulatore SSB, sentireste solo dei rumori incomprensibili. L'assenza di portante, nella trasmissione DSB-SC, rende inutile la demodulazione dell'inviluppo tramite il classico rilevatore usato nella normale modulazione di ampiezza, ma se siete già in possesso di un ricetrasmettitore amatoriale non avrete difficoltà a sintonizzarvi sui 14.318 MHz e ascoltare cosa propaga nell'etere il vostro piccolo trasmettitore. ❑

CODICE MIP 2756967

Pulisco qualsiasi tipo di pavimento

Arrivo negli angoli e nelle zone più difficili da raggiungere

Passo sotto i mobili

Aspiro sporco, polvere, briciole e peli di animali

Sono Roomba®, il robot aspirapolvere che pulisce da solo

iRobot Corporation 2009. Tutti i diritti riservati. iRobot e Roomba sono marchi depositati di iRobot Corporation.

iRobot®

Making Robots Work For You

Pulisco meglio di chiunque, perché sono un robot. Tutti i miei movimenti sono calcolati 64 volte al secondo. Grazie al mio speciale sensore che rileva lo sporco, passo tutto il pavimento più volte e quando è perfettamente pulito, vado alla stanza successiva* oppure rientro alla Home Base per ricaricarmi.
Posso essere programmato** facilmente per pulire dove e quando vuoi tu. Riconosco e pulisco qualsiasi tipo di pavimento, dalle piastrelle ai tappeti, dal parquet alla moquette. Aspiro qualsiasi tipo di sporco: polvere, briciole e peli di animali. Sono così bravo che possiedo molti brevetti e ho vinto premi di design e ingegneria in tutto il mondo. Gli ingegneri che mi hanno progettato hanno costruito anche i miei amici che lavorano per la NASA, ma io preferisco lavorare per le oltre 3 milioni di famiglie che contano su di me. Anche tu, come loro, mi darai un nome (ah, questi umani!). Ma io ho già un nome, **mi chiamo Roomba**.

*Modelli 560, 562, 580 (modelli con Lighthouse in dotazione) **Modelli 555, 560, 562, 580

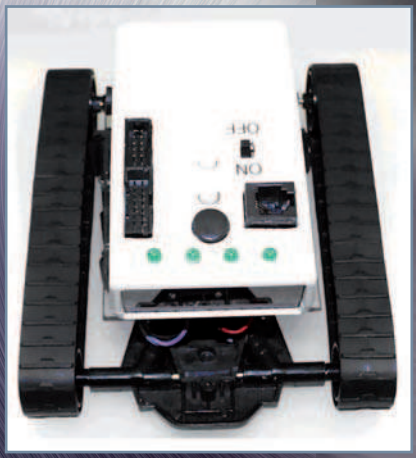
GARANZIA UFFICIALE ITALIA 2 ANNI iRobot con Nital

CODICE MIP 2757593

www.irobot.it

I prodotti iRobot sono distribuiti in Italia da Nital

ROBOTZone



96 ROBOT KIT L'HARDWARE

Progettare piccoli robot domestici è certamente un'attività gratificante che può riempire di gioia qualsiasi appassionato.
Come ogni attività umana, e in modo particolare un lavoro di progettazione integrata hardware e software, è necessario definire, in modo chiaro e preciso, le diverse fasi di lavoro: solo in questo modo possiamo raggiungere il nostro obiettivo

di Francesco Pentella

104 THREAD FOCUS

La tua voce dalla rete

ROBOT KIT *L'hardware*

(prima prima)

Progettare piccoli robot domestici è certamente un'attività gratificante che può riempire di gioia qualsiasi appassionato. Come ogni attività umana, e in modo particolare un lavoro di progettazione integrata hardware e software, è necessario definire, in modo chiaro e preciso, le diverse fasi di lavoro: solo in questo modo possiamo raggiungere il nostro obiettivo

La realizzazione di un robot, anche semplice, richiede un certo lavoro che coinvolge gli aspetti legati al design hardware fino a coprire anche il software di basso livello (o firmware). Per raggiungere il nostro obiettivo occorre definire a tavolino tutti gli aspetti, ma quali? Questo articolo propone la definizione di un piccolo robot in grado di compiere movimenti autonomi. Per intraprendere questo percorso utilizziamo un piccolo progetto messo a punto diversi anni fa: il RoCK (o Robot Conversion Kit), consultabile dal sito www.wirz.com/rock che fornisce ulteriori informazioni su questo oggetto. Il RoCK è un modulo, a basso costo, che permette di trasformare un giocattolo, per esempio una macchinina, in un robot mobile. La **figura 1** e la **figura 2** mettono in evidenza le due diverse soluzioni. Nella **figura 1** vediamo l'uso del RoCK su una macchina, o su un altro dispositivo mobile. In questo articolo affronteremo il design dell'hardware, mentre nel prossimo ci occuperemo della parte software del progetto.

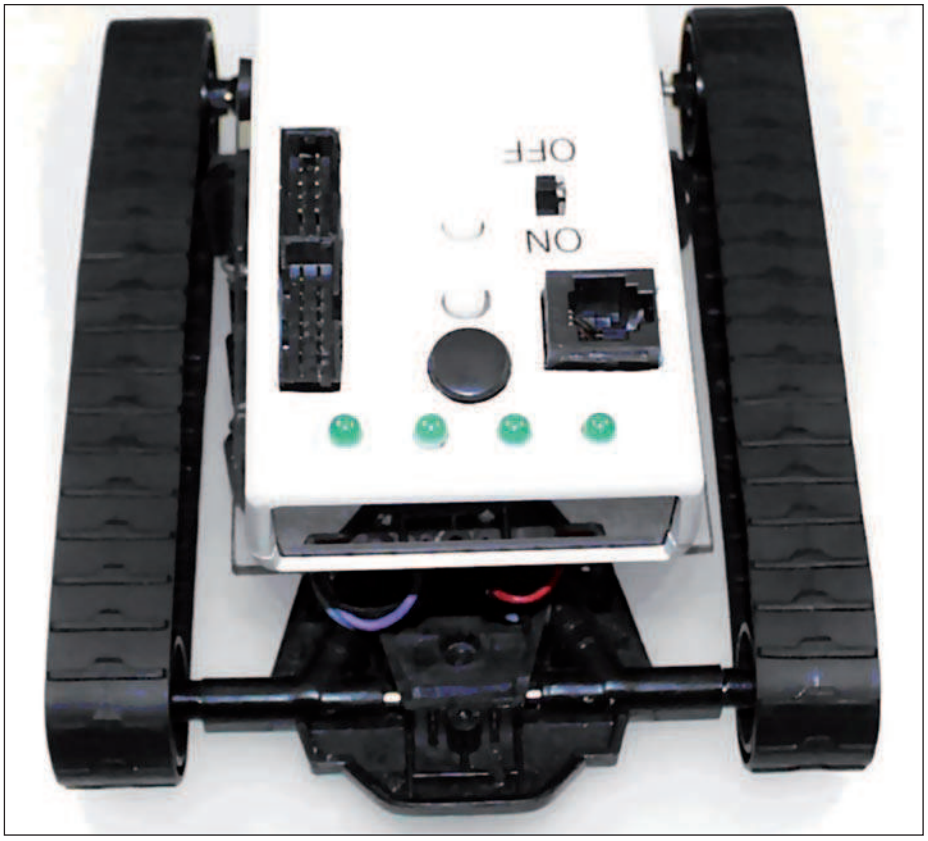


Figura 1: il nostro robot all'opera.

OVERVIEW DEL SISTEMA

La **figura 3** mostra lo schema top-level del sistema. Da questo diagramma notiamo i diversi sensori, come ad esempio quelli IR (infrarossi) per la rilevazione degli ostacoli. I sensori di luce permettono al nostro robot di seguire una fonte d'illuminazione o anche di evitare la stessa per cercare zone oscure o in ombra. Il sistema per la rilevazione degli ostacoli è basato su quattro diodi IR, due emettitori e due ricevitori, che permettono al robot di evitare collisioni. Il principio di funzionamento di un sistema del genere è presto spiegato. I due ricevitori utilizzano una frequenza tipica di 38.5 kHz, mentre per regolare la sensibilità si ricorre a un potenziometro, vale a dire con il potenziometro si regola l'assorbimento (quantità di luce) dei due emettitori. Se un ostacolo si trova nel cono di luce generato dal LED IR sinistro, questo ne riflette la luce verso il ricevitore IR e quindi viene indicata la presenza dal lato sinistro; al contrario, se l'oggetto si trova dal lato destro e riflette la luce del LED IR destro, questo sarà rilevato come presente dal lato destro. In caso di collisione, la sezione del circuito predisposto a questa evenienza metterà in allerta il robot e in questo modo il software attiverà le varie azioni correttive predisposte allo scopo. Il driver del motore fornisce poi la necessaria potenza per pilotare correttamente la macchina. È possibile configu-

rare il robot attraverso diversi interruttori presenti e il potenziometro, in questo modo l'utilizzatore è in grado di regolare diversi parametri.
In questa realizzazione è anche presente una sorta di monitoraggio del valore della tensione erogata dalla batteria, una proposta interessante perché fornisce all'utilizzatore le necessarie informazioni per decidere la sostituzione della stessa. Inoltre, grazie alla presenza di quattro LED e di un cicalino, il robot è in grado di segnalare alcune condizioni di funzionamento. Queste condizioni possono essere definite nel firmware attraverso opportune configurazioni del software. La presenza di un'interfaccia seriale, poi, garantisce al sistema flessibilità in fase di aggiornamento del firmware (presente nella memoria interna microprocessore). Il sistema permette di collegare sensori addizionali o attuatori utilizzando un *expansion header*, in questo modo diventa perfettamente scalabile. La parte intelligente del sistema è gestita da un microprocessore della famiglia Atmel. La **tabella 1** mostra i diversi segnali coinvolti e le loro relazioni con i pin del processore.

IL PROCESSORE

Il blocco CPU è messo in risalto dalla **figura 4**. Il nostro progetto utilizza il modello Atmel AVR AT90S7535, ma può essere sostituito, perché in via di obsolescenza, dal modello ATmega8535. L'uso di un microcontrollore permette di risparmiare spazio e riduce i costi di realizzazione.
Il modello dispone di una serie di canali per supportare la funzionalità ADC (*analog to digital converter*) utilizzata per monitorare i sensori. Il processore fornisce anche diversi timer che trovano la loro naturale applicazione nella funzionalità di PWM (*pulse width modulation*) per il pilotaggio di motori, nella generazione dei toni per il cicalino, o per la necessaria frequenza di 38 kHz utilizzata dal sottosistema per la rilevazione degli ostacoli (IR). L'AVR dispone anche di diversi comparatori per il controllo dell'*over-current* del driver dei motori. Il processore dispone anche di diverse *facility* che rendono

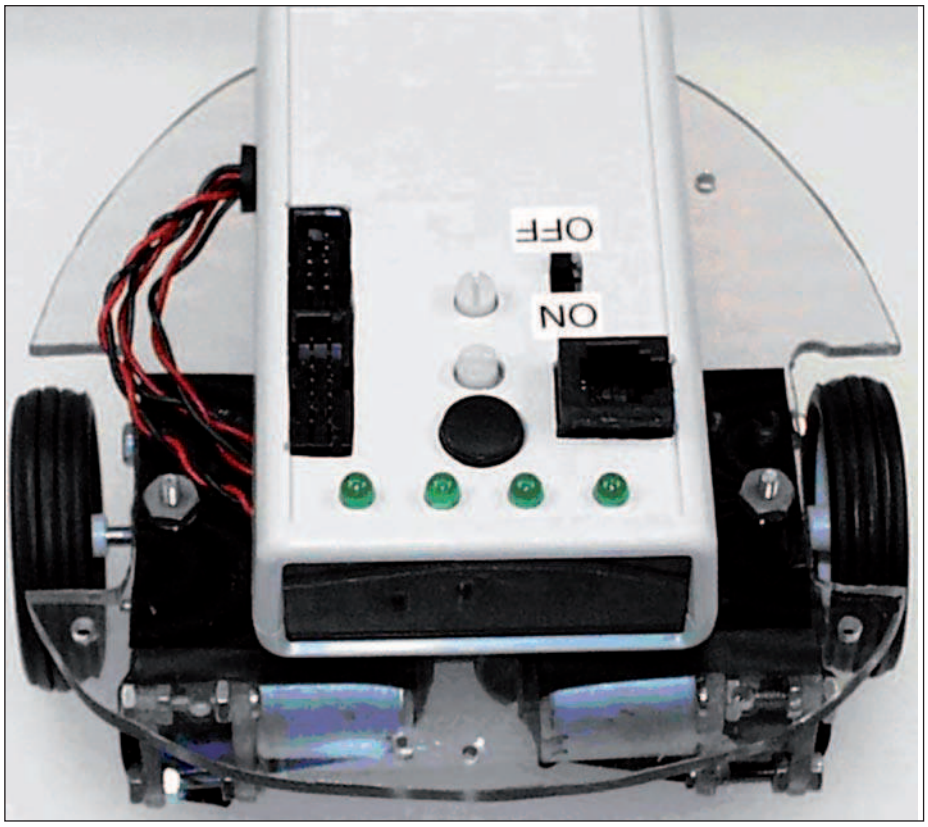


Figura 2: un'altra applicazione.

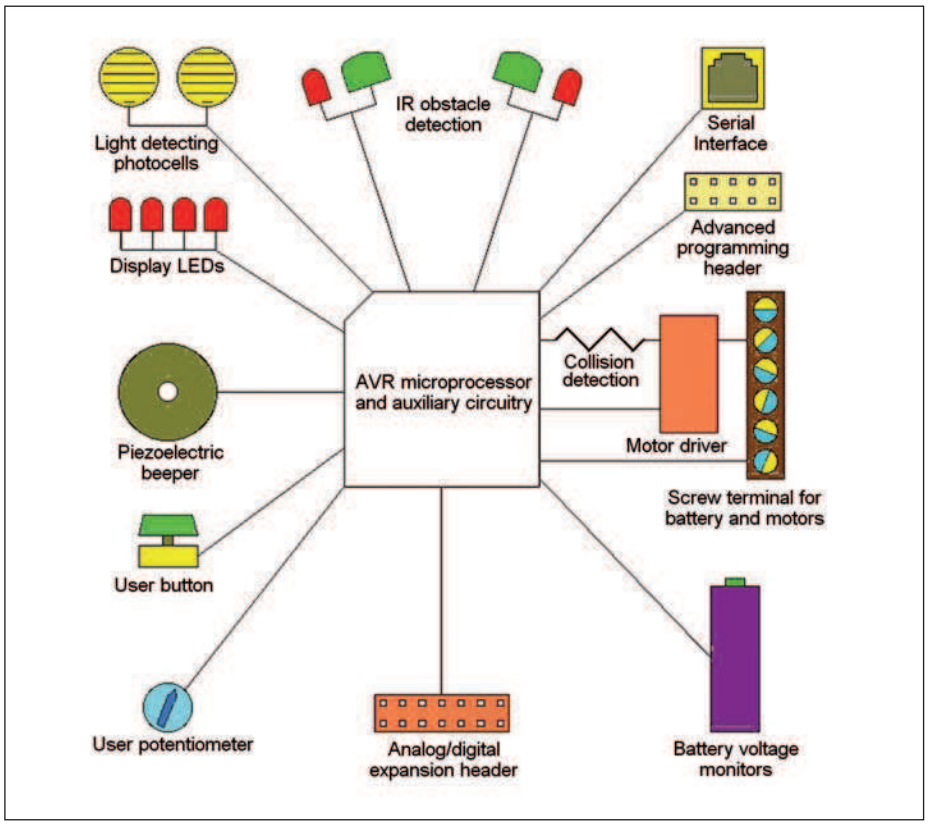


Figura 3: top level.

Tabella 1 – Funzionalità e segnali coinvolti

SISTEMA	PORTS/SIGNALS ATMEL	DESCRIZIONE
Interfacce On-board		
Power switch	Vcc/Batt+	DPDT switch enables logic and motor power
Display	PC0-3	Quattro display per evidenziare lo stato del robot
Cicalino	PD6	Segnali pilotato ad interrupt. Il cicalino rIproduce toni con frequenza arbitraria
Potenziometro a disposizione dell'utente	PA5 (ADC5)	Attraverso il potenziometro è possibile regolare e configurare la nostra applicazione
Tastini per l'utilizzatore	PA4	Consente la configurazione della nostra applicazione
Interfaccia seriale	PD0,1 (RXD,TXD)	L'uso dell'interfaccia RS-232 permette di collegarsi a un PC per sessioni di lavoro
Rilevazione degli ostacoli		
Left IR detector	PD2	38 kHz IR receiver
Right IR detector	PD3	38 kHz IR receiver
IR emitters	PD7 (OC2)	La generazione dei 38 kHz è fatta direttamente dal processore
Sensori di luce		
Left photo cell	PA0 (ADC0)	La rilevazione delle fonti di illuminazione è fatta attraverso degli ingressi analogici
Right photo cell	PA1 (ADC1)	Collision detection
Motor current	PB2 (AIN0)	Comparatore analogico. Attraverso questa funzionalità è possibile fare una misura per rilevare le collisioni
Current set	PB3 (AIN1)	
Battery monitors		
Motor power	PA2 (ADC2)	Monitor for motor supply battery voltage
Logic power	PA3 (ACD3)	Monitor for logic supply battery voltage
Motor output		
Motor PWM	PD4,5 (OC1B,A)	La generazione del segnale PWM è svolta interamente dal processore
Motor direction	PB0,1	
IO addizionale		
User digital	PC0-7	Otto I/O digitali disponibili all'utente
User analog	PA6,7 (ADC6,7)	Due canali analogici disponibili

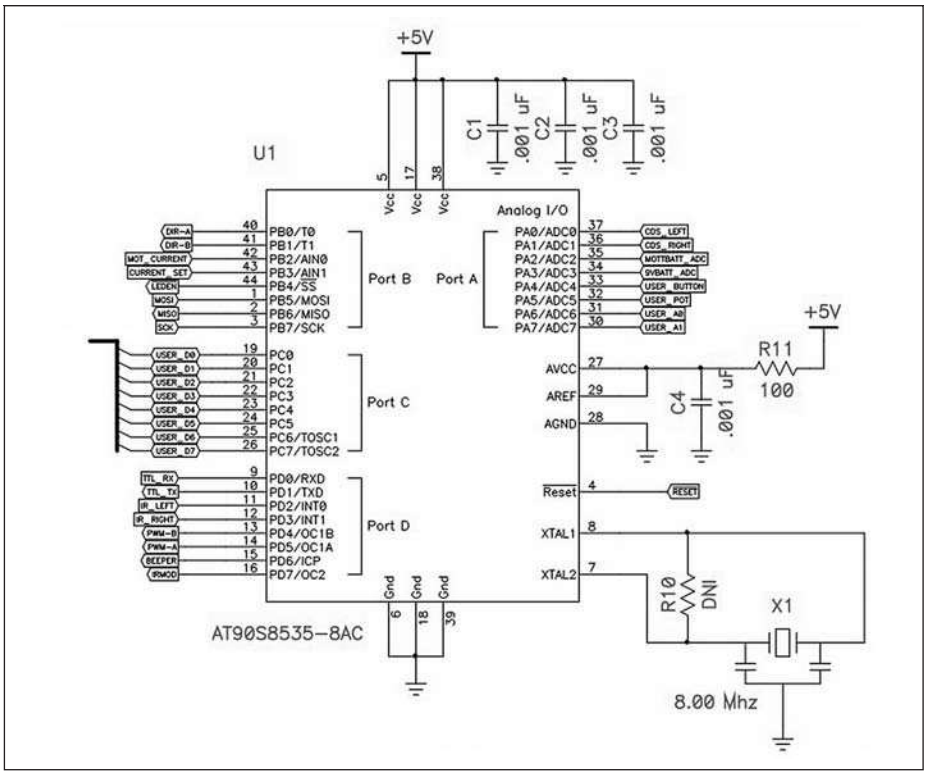


Figura 4: modulo processore.

il lavoro del programmatore abbastanza agevole. Per esempio, la presenza delle seriali *on chip* può essere senz'altro utile per collegare il processore a un PC per le seguire senza problemi sessioni di programmazione e il *monitoring* di *high-level*. Oltre alla presenza di segnali analogici, il processore permette anche la gestione delle linee di I/O digitale. L'uso delle linee digitali è di fondamentale importanza per la gestione dei LED e per il tastino *user button*.

Il modello AT90S8535 dispone di 8 KB di memoria flash utilizzata per il programma di gestione del sistema, di 512 KB di memoria SRAM e di 512 KB di memoria EEPROM. La memoria presente nel processore è utilizzata per ospitare il programma di funzionamento del robot o dei vari programmi utente. Per la gestione della memoria non sono necessari altri dispositivi hardware perché il blocco funzionale è praticamente gestito direttamente dal processore senza la necessità di circuiteria addizionale. La fre-

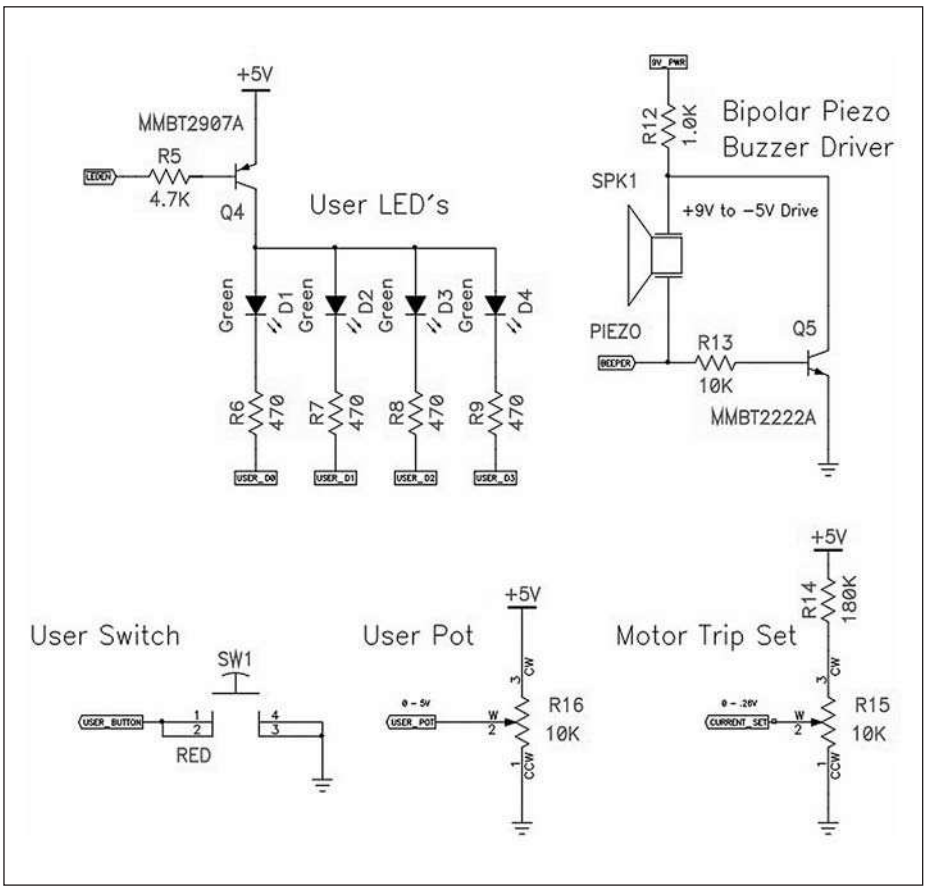


Figura 5: on board user interface circuit.

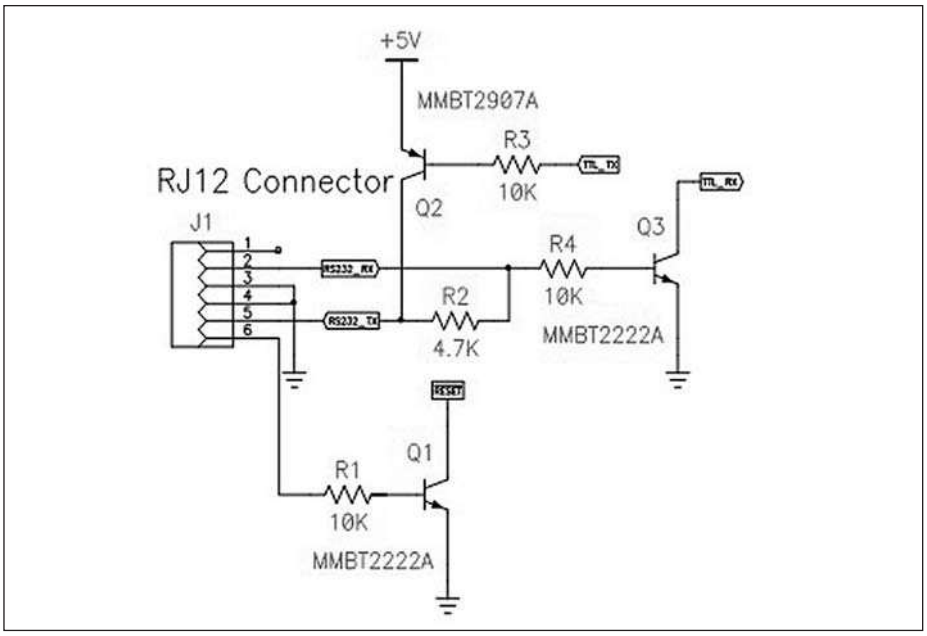


Figura 6: Rs-232.

quenza di funzionamento del microprocessore è assicurata dalla presenza di un oscillatore di 8 MHz di tipo ceramico presente nel circuito.

INTERFACCE ON-BOARD

Quattro linee di I/O sono configurate come output (**figura 5**), in questo modo è possibile collegare al robot una serie di LED o display. Questi LED potrebbero essere utilizzati per indicare lo stato del robot. La linea attiva bassa, LED enable line o LEDEN, rende possibile il multiplexer delle linee che pilotano i LED. Il cicalino è connesso alla linea PD6 di Atmel. Per generare un'onda quadra per il cicalino si utilizza una ISR (*Interrupt Service Routine*) agganciata al *Timer/Counter 0* per fare così il toggle del pin PD6. Il tempo che intercorre tra due interrupt corrisponde al mezzo periodo, H, del suono emesso dal cicalino. Il software utilizzato dal robot è regolato per un tick di 8 microsencondi. In questo modo:

$$f = \frac{1}{2H8 \cdot 10^{-6}} \text{ o } f = 62,500 / H.$$

Il Timer/Counter 0 è un registro a 8 bit con valori tra 0 e 255. La più piccola frequenza che il cicalino può produrre è di f = 62,500/255 = 245 Hz, mentre la maggiore frequenza (dove H è uguale a 1) è di 62.5KHz. Il robot restringe l'insieme dei toni riprodotti dal cicalino, vale a dire che, per un tono con una frequenza determinata, f_0, l'ennesimo tono della scala è dato dalla relazione

$$f_n = f_0 2^{n/12}$$

Il robot utilizza una frequenza di 532 Hz per f_0. Nella definizione della scala solo i valori interi possono essere considerati e memorizzati. Per evitare problemi con il volume del cicalino è possibile utilizzare uno pseudo driver H-bridge per creare una 14Vpp, il tutto per mezzo di un transistor esterno e due resistenze (si veda la **figura 5**).
Per selezionare un modulo software presente nella memoria del processore, e quindi una differente modalità di funzionamento, l'utente deve premere il tastino definito come *user button* e regolare il potenziometro. Il potenziometro è collegato al canale cinque della linea ADC e il tastino al pin PA4 del processore (entrambi appartengono alla porta A del processore). Si ricorda che ogni pin della porta A è configurabile autonomamente per applicazioni digitali o analogiche. La figura 6 mostra, poi, il circuito che implementa una RS-232 di tipo half-duplex. Un'interfaccia seriale permette la comunicazione con

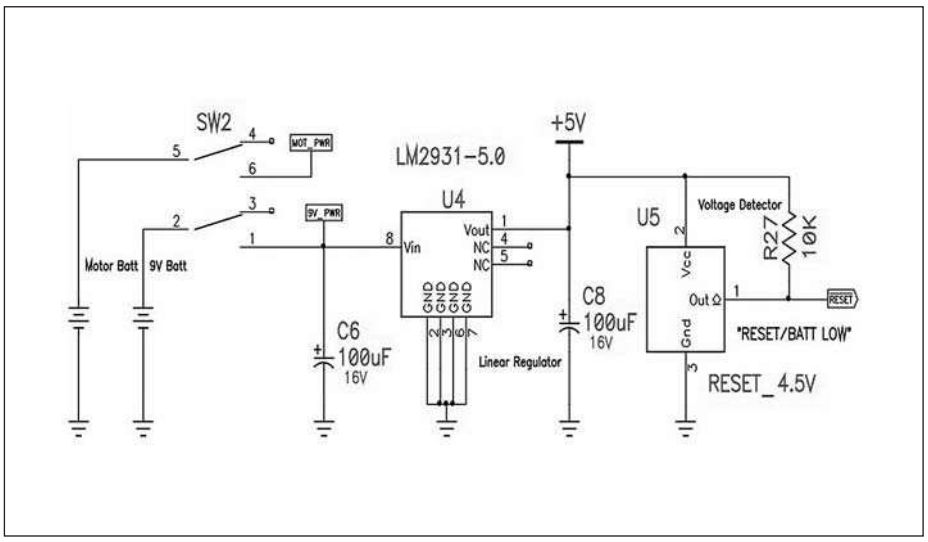


Figura 7: Power supply and reset.

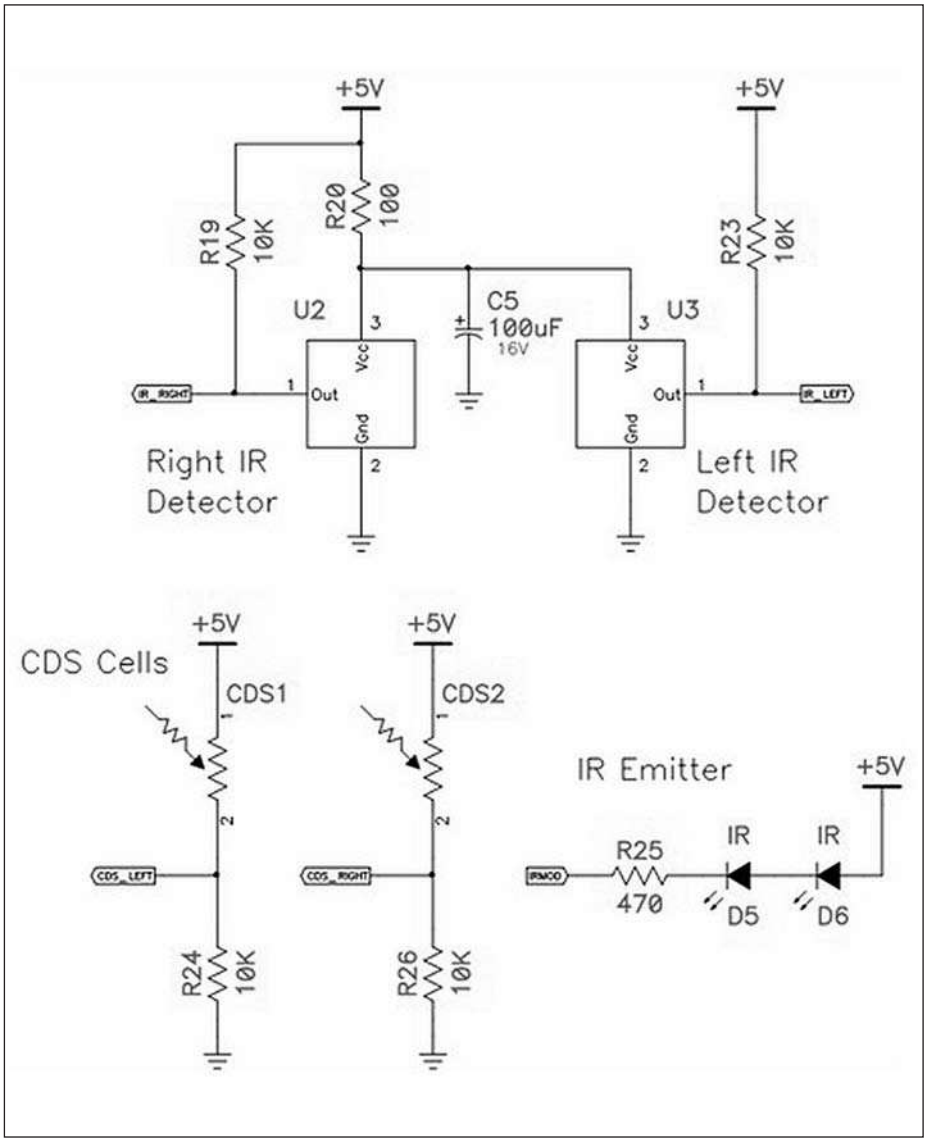


Figura 8: IR detector, IR Emitter, and light sensing circuits.

un host per instaurare, per esempio, sessioni di test con la conseguente riduzione di costi e spazio di board evitando l'uso di blocchi funzionali addizionali.
La seriale funziona a 9600. Il valore del baud rate viene modificato attraverso il registro UBRR. La velocità della seriale è ricavata dalla formula:

$$BAUD = \frac{f_{CK}}{16(UBRR+1)}$$

dove fCK è la frequenza del clock
L'uso del registro UBR, a un valore di 51, fissa i valori minimi e massimi del baud. Così avremo:

$$BAUD_{max} = \frac{8.088MHz}{16(51+1)} = 9,721bps$$

$$BAUD_{min} = \frac{7.912MHz}{16(51+1)} = 9,510bps$$

I valori di 8.088 MHz e 7.912 MHz tengono conto della tolleranza dell'oscillatore. Il baud rate prodotto anche in questo caso oscilla tra 1.3% e -0.9% della sua velocità ideale di 9600 bps.

POWER SUPPLY & RESET

Il blocco funzionale della sezione alimentazione e reset è mostrato nella **figura 7**. La prima versione del robot utilizzava un comune circuito integrato 7805 in un package TO-92. È stata scelta una batteria da 9 V per pilotare tutta la parte logica. A oggi possiamo senz'altro pensare di impiegare i più moderni regolatori di tensione per ottenere bassi valori di dropout rispetto al tradizionale 7805. Grazie a questo otteniamo perciò un vantaggio immediato, vale a dire l'aumento della vita media della batteria. L'aspetto negativo invece è il diretto aumento dei costi; infatti, la maggior parte dei regolatori di tensioni con basso dropout sono più costosi di un tradizionale 7805.
Probabilmente, cercando tra i vari costruttori, possiamo trovare anche delle soluzioni con un prezzo, magari non inferiore, ma certamente comparabile. Per esempio, il dispositivo della National

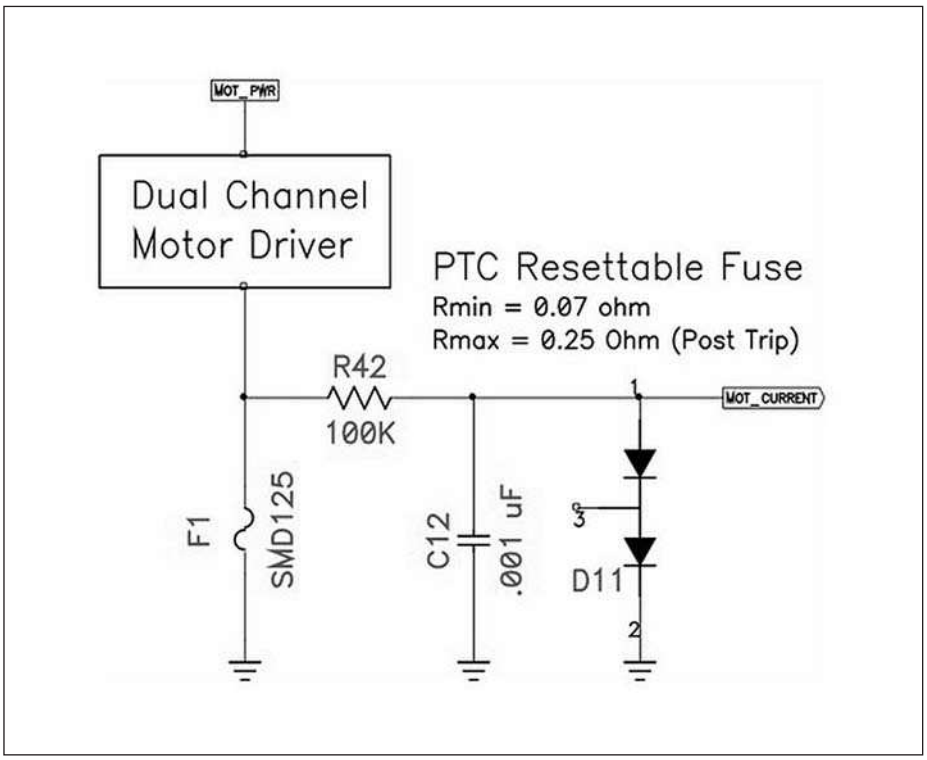


Figura 9: Collision detection and over-current protection circuits.

Semiconductor LM2931 utilizza solo un condensatore elettrolitico standard per filtrare l'uscita.

LIGHT SENSORS

Questa parte è messa in evidenza dalla **figura 8**. In questa realizzazione si utilizzano elementi foto-resistivi di tipo CdS (cadmium sulfide). Il tempo associato alle celle CdS può eccedere i 35 ms, ma questi tempi sono in linea con le nostre esigenze.

LA RILEVAZIONE DEGLI OSTACOLI

Il progetto prevede che il sistema deputato alla rilevazione degli ostacoli debba essere composto da due diodi emettitori IR connessi in serie e due ricevitori (vedi **figura 9**). Dato che la corrente richiesta è abbastanza bassa, i diodi emettitori sono pilotati dalle linee del microcontrollore PD7 (OC2). I segnali attivi bassi dai diodi IR sono collegati alla linea PD2 e PD3, rispettivamente, dei ricevitori di sinistra e destra. I dispositivi IR ricevitori sono gli stessi già utilizzati per rilevare in remoto i segnali di controllo di un apparecchio televisivo o di un altro dispositivo elettronico di consumo. Per funzionare correttamente, il diodo IR receiver deve funzionare con un segnale infrarosso a 38 kHz modulato attorno a 1 kHz. Il processore utilizza il Timer/Counter 2 per generare il segnale di 38 kHz per mezzo del conteggio impostato dal software, conta un numero corrispondente di mezzi periodi di una frequenza di 38 kHz.
Quando il contatore incontra questo valore, la linea PD7 si mette in toggle e il contatore resetta per ripartire nuovamente. La connessione è commutata ogni 500 microsecondi. È compito del software del processore attivare e disattivare, alternativamente, la connessione tra Timer/Counter 2 e la linea PD7.

MOTOR DRIVER

La scelta del tipo di motore da utilizzare per un'applicazione di questo tipo rappresenta la difficoltà maggiore. Per la nostra realizzazione, il motore dovrebbe avere le seguenti caratteristiche:

- Dual channel con massima corrente continua di 500 mA per canale;
- tensione di ingresso tra 3-15 VDC;
- possibilità di essere interfacciato con un processore AVR con PWM;
- possibilità di sfruttare l'over *current protection* e il *bump sensing*.

Sicuramente il modello che incontra il maggior favore è Unitrode/STMicroelectronics L293D. Il dispositivo L293D può essere configurato come un duale driver H-bridge ed è disponibile a un costo contenuto e proposto in package DIP16. Inoltre, il componente per essere utilizzato correttamente richiede solo un apporto minimo di componenti a supporto. Il componente presenta però anche degli svantaggi. Innanzitutto, la durata di una batteria alcalina a 9 V è di alcune ore. Esistono in commercio altri componenti di questo tipo che forniscono prestazioni decisamente migliori. Per esempio, il Vishay Siliconix Si9986, anche se il prezzo risulta maggiore. Esiste anche un'alternativa, vale a dire mettere a punto noi stessi un driver per motori con dei transistor. Un design utilizzando dei transistor bipolari fornisce sicuramente le risposte alle nostre esigenze. Inoltre, un design di questo tipo costerebbe di meno oltre a evitare le diverse complicazioni di un disegno con MOSFET.
Il driver del nostro robot è protetto dalle escursioni di corrente eccessive per mezzo di un PTC (*positive temperature coefficient*) a 1.25 A. La corrente del motore è monitorata per rilevare collisioni. Quando il robot incontra un ostacolo, il motore incrementa il flusso di corrente in risposta al carico addizionale che il robot richiede per via della presenza di un ostacolo. Monitorando la corrente, il robot è in grado di rilevare quando incontra un ostacolo sulla sua strada. Il diodo D11 ha la funzione di limitare la tensione presente agli ingressi del processore. La resistenza R42 e la capacità C12 aiutano il filtraggio di spike di corrente.
Il progetto può anche essere modificato per inserire motori con diversi requisiti di corrente.

I/O AUSILIARIO

È possibile prevedere nuovi canali di I/O intervenendo sull'I/O header ausiliario. Attraverso PA6 e PA7 o da PC0 a PC7 sono disponibili, rispettivamente, due ingressi analogici o digitali oppure. Le linee da PCO a PC3 sono condivise con i LED. ❑

CODICE MIP 2757083

www.baronerosso.net

mappa del sito contatta

10 settembre 09, 16:28 #1 (permalink) Top

jeky84
User

interruttore mosfet

Buon pomeriggio a tutti
sono nuovo del forum e colgo l'occasione per presentarmi e per porgervi una domanda:
io ho un micro che invia dai piedini d'uscita un segnale logico alto(5V) o basso(0V), devo connetterlo a dei mosfet di modo tale che questi mi facciano da interruttore per la centralina(cioè quando il micro invia segale a livello logico alto il mosfet si deve " chiudere" quando invia seganle a livello logico basso il mosfet si deve aprire) la quale prenderà come alimentazione una 9V da banco.
considerate che non mi interessa il valore di corrente che arriva alla centralina...anzi meno è meglio è visto che mi serve solo che arrivi l'impulso alla centralina per abilitarne l'utilizzo
vorrei sapere il mosfet che caratteristiche deve avere per permettermi di fare questo lavoro e se occorre un'opportuna rete di impedenze intorno ad esso
Ringrzio tutti anticipatamente
Saluti

10 settembre 09, 17:19 #2 (permalink) Top

SoldatoSemplice
User

Se vuoi proprio usare un mosfet dovresti trovarne uno che abbia una Vgs-on compatibile con i livelli TTL, ma visto che sono abbastanza rari e sicuramente costano di più relativamente ad un comune mosfet, devi prima amplificare di un pò il segnale di tensione che esce dal tuo micro e l'uscita dell'amplificatore la mandi al gate del tuo mosfet, in questa applicazione io ti consiglio di usare un amplificatore operazionale in configurazione di amplificatore non invertente, in questo modo, complice il fatto che l'impedenza d'ingresso di un OPAMP è altissima e quella d'uscita è molto più bassa realizzi anche un primitivo adattamento d'impedenza.

In alternativa invece dell'OPAMP puoi usare un semplice BJT da 20 centesimi di tipo NPN però in questo caso ti devi assicurare che il tuo micro in uscita dia una corrente sufficiente a polarizzare la base del BJT stesso che userai come amplificatore prima del mosfet.

10 settembre 09, 17:50 #3 (permalink) Top

Naraj
User

Perchè non usare un semplice transistor ?

10 settembre 09, 19:21 #4 (permalink) Top

marcosinatti
User

Azz Soldatosemplice impara velocemente forum cnc italia • Leggi argomento - Per uno stadio di potenza, meglio saturazione o la classe A?

ahah screrzo... riporto comunque la risposta che avevo dato in quel forum

Il mosfet si pilota in tensione tra G ed S, se guardi i datasheet vedrai che con una tensione di 7-8V già il mosfet lavora in saturazione a la sua RDSon è molto bassa anche con correnti elevate.
Non pilotarlo con tensioni sopra i 12V in quanto interviene la protezione del gate interna fatta con due zener che ti porterebbe solo spreco di corrente di pilotagio e surriscaldamento del mosfet stesso.
Ricorda che il mosfet sul gate non ha bisogno di grosse correnti però ha una discreta capacità che potrebbe romperti le balle ad alte frequenze, quindi il circuito che lo pilota deve essere a bassa impedenza, possono andare bene 2-3 porte cmos in parallelo oppure un transistor npn configurato ad emettitore comune e con una resistenza verso il positivo da 1k.
Se usi il mosfet con basse correnti di uscita e non ti interessa quindi saturarlo al minimo dellla RDSon puoi pilotarlo con i 5V del micro direttamente, mettendo in serie una resistenza da 180-220Ohm.

MARCO
T-rex 450SE Carbon Metal - SM450 - HK450 --- SM-PILOT By SMTech [The Stabilizer]

10 settembre 09, 22:02 #5 (permalink) Top

SoldatoSemplice
User

Questo non lo sapevo, forse hai avuto un idea migliore tu.

11 settembre 09, 09:04 #6 (permalink) Top

jeky84
User

grazie mille ragazzi..era proprio la risposta che cercavo...infatti a me non interessa saturare il mosfet perchè ho basse correnti in uscita...quello che vorrei sapere è il tipo di mosfet che devo utilizzare, se un P o un N, e inoltre se lo piloto con i 5V del micro e il segnale di alimentazione che deve andare alla centralina è di 9V quale mosfet potrei usare...grazie mille a tutti!!

12 settembre 09, 12:26 #7 (permalink) Top

marcosinatti
User

Se i tuoi due circuiti hanno lo 0V in comune e ti va bene che chiuda verso massa va bene un canale N qualsiasi, viste le basse correnti e tensioni in gioco, con google ne trovi quanti ne vuoi, io spesso uso il sito della rscomponents anche per individuare i componenti, è più veloce.

MARCO
T-rex 450SE Carbon Metal - SM450 - HK450 --- SM-PILOT By SMTech [The Stabilizer]

NNTP
Newsgroup Gateway

Username Username Ricordami?
Password Entra
Google Ricerca personalizzata Cerca

NNTP.IT » Forum » Newsgroups it.hobby.* » it.hobby.elettronica » spettro onda quadra

Strumenti discussione

23-11-2009, 21.09.00 #1

brown_fox
Messaggi: n/a

spettro onda quadra

Che livello hanno le armoniche (dispari) di un'onda quadra?
In termini molto pratici, se invio a un analizzatore
di spettro un'onda quadra a 10 MHz e la fondamentale
è a 20 dBm, che ampiezza avrà la terza armonica? E le
sucessive, espresse in dBm?

23-11-2009, 22.06.39 #2

LaCariatide
Messaggi: n/a

Re: spettro onda quadra

Le armoniche (dispari) di una onda quadra hanno una ampiezza che
diminuisce come 1/n, dove n e'l'ordine della armonica.
Per cui la terza armonica ha una ampiezza pari ad 1/3 della
fondamentale, ovvero 20*log10(1/3)=-9.5 dBc .
In pratica questo vale fino a frequenze comparabili col tempo di
salita dei fronti dell'onda quadra.

Saluti,
LaCar.

24-11-2009, 14.16.36 #3

brown_fox
Messaggi: n/a

Re: spettro onda quadra

Grazie della risposta, ma perchè 20*log...? Se stiamo
lavorando con potenze, come nel caso dell'analizzatore
di spettro, non dovremmo moltiplicare per 10?

24-11-2009, 14.30.57 #4

LaCariatide
Messaggi: n/a

Re: spettro onda quadra

No, i decibel sono sempre decibel, che di ampiezza o potenza si parli.
Poiche' l' *ampiezza* delle armoniche diminuisce come 1/n e'corretto
usare il fattore 20, ovvero 20*log10(1/n).
O per vederla da un altro punto di vista, se l' *ampiezza* diminuisce
come 1/n, la *potenza* diminuisce come 1/(n^2) per cui puoi scrivere
10*log10(1/(n^2)) che per le proprieta'dei logaritmi e'uguale a
20*log10(1/n).

Saluti,
LaCar.

26-11-2009, 10.16.47 #5

giorgiomontaguti
Messaggi: n/a

Re: spettro onda quadra

Ciao
-9.5 dB significa contemporaneamente una riduzione
in tensionei a un terzo e una riduzione in potenza a un nono,
semplicemente perche' P=(V^2)/R.

Ciao Giorgio

--
non sono ancora SANto per e-mail

FiserTek
Robotica sperimentale

www.FiserTek.it
forum di supporto al sito

Indice » Elettronica » elettronica digitale — Tutti gli orari sono UTC +1 ora [ora legale]

Punta del saldatore

Moderatori: fiser, tls-service

Stampa pagina — Precedente | Successivo

Autore	Messaggio
ferrettip	**Oggetto del messaggio:** Punta del saldatore — **Inviato:** 03/09/2009, 22:59 Buona a tutti, ho un grosso problema, probabilmente causato dalla mia inesperienza: le punte del saldatore mi si "ossidano" dopo pochi utilizzi. Utilizzo il filo con il 99,3% di stagno e 0,7% di rame, anima con flussante e temperatura di fusione circa 230 °C. Il saldatore lavora con potenza regolabile a 15W in condizioni normali, se spingo un pulsante aumenta a 30W. All'inizio, durante e alla fine delle operazioni di saldatura immergo la punta nella pasta salda. Cosa sbaglio? Qualcuno può aiutarmi e darmi dei suggerimenti per evitare di spendere € 7,00 a settimana? Grazie
fiser	**Oggetto del messaggio:** Re: Punta del saldatore — **Inviato:** 03/09/2009, 23:09 probabilmente hai pulito la punta raschiandola con qualcosa. io uso pulire con flussante e subito dopo una passata su spugna umida senza mai usare superfici abrasive. in questo modo dura secoli. l'unico modo per rovinare la punta è appunto raschiarla, si toglie la pellicola superficiale e buona notte.
ferrettip	**Oggetto del messaggio:** Re: Punta del saldatore — **Inviato:** 03/09/2009, 23:13 Fiser so che la punta è nichelata e che raschiandola si toglie questo sottile strato di nichel, pertanto non la tocco con niente! Al contrario la tua risposta mi ha dato da pensare al fatto che io alla fine delle operazioni la immergo nella pasta salda e non la pulisco con la spugna bagnata. Secondo te potrebbe essere che la pasta che rimane depositata sulla punta la ossida...?
Ale72	**Oggetto del messaggio:** Re: Punta del saldatore — **Inviato:** 03/09/2009, 23:17 Beh, io ancora uso stagno/piombo 60/40... o magari con un 2% di argento... ne ho talmente tanto! Immagino che questa tua lega sia per le nuove norme che vietano il piombo... ma ancora non ho affrontato il problema. Quello che sapevo però era che le nuove leghe richiedevano una temperatura leggermente superiore e non inferiore... comunque, se ti si fonde bene, nessun problema! Le punte dei saldatori buoni hanno un trattamento superficiale chiamato long-life (lunga vita appunto) che non espone l'anima interna in rame della punta agli agenti esterni. Se le cattive abitudini ti portano ad utilizzare la punta del saldatore magari per fondere plastica (AAARGHHH!!! Sacrilegio!! è un saldatore! Serve SOLO a fare il suo lavoro!!!) magari avrai anche la cattiva abitudine di passarci spugnette abrasive, lana d'acciaio o peggio tela smeriglio per ripulirlo! In quel modo si asporta lo strato protettivo superficiale della punta e viene fuori il rame nudo che a contatto con la lega di saldatura, la pasta disossidante e l'ossigeno fa' delle belle crostone nere! Poi, se invece il tuo saldatore è un Philips... beh! Allora non puoi farci niente anche le punte Long-life non servono a nulla. Se ancora le tue punte sono in rame nudo è impossibile eliminare il problema. Un consiglio: -Pulisci sempre le punte con l'apposita spugnetta !
fiser	**Oggetto del messaggio:** Re: Punta del saldatore — **Inviato:** 03/09/2009, 23:18 la pasta altro non è che acido cloridrico reso pastoso con non ricordo cosa. (era una ricetta facile da fare in casa... mi pare canfora o qualcosa del genere) quindi è comunque aggressivo essendo appunto disossidante nei confronti del rame. il flussante va usato solo appena prima di pulire la punta sulla spugna bagnata. le temperature di funzionamento mi tengo sempre intorno ai 190 gradi... max 200, mai di più che già sono tantissimi.
fiser	**Oggetto del messaggio:** Re: Punta del saldatore — **Inviato:** 03/09/2009, 23:20 ovviamente si presuppone di parlare di un saldatore serio. se si prende in considerazione quello che chiamo "cacciavite" elettrico è meglio buttalo al secchio
Ale72	**Oggetto del messaggio:** Re: Punta del saldatore — **Inviato:** 03/09/2009, 23:26 ...non ho capito se c'è un numero max di caratteri o se c'è stato un problema... comunque riscrivo il seguito del mio ultimo messaggio qui: Dicevo: -pulisci la punta solo con l'aposita spugnetta ben bagnata e strizzata; -NON pulire la punta al termine delle operazioni di saldatura! Bagnala bene con dello stagno nuovo (in modo che gli rimanga attaccato) e spegni il saldatore. Quello stagno sarà un eccellente protettivo! -In linea di massima io NON uso disossidante! (quello che si trova in commercio è di qulità peggiore rispetto alla quantità, peraltro giusta, contenuta nella lega saldante stessa) - Se saldi con una certa assiduità ti consiglio di valutare l'acquisto di una stazioncina a bassa tensione della Weller! Anche una stupida...e vedrai che il tuo modo di lavorare cambierà! Ciao. Alessandro.

Visualizza ultimi messaggi: Tutti i messaggi — Ordina per Ora di invio — Crescente — Vai

Roboitalia
www.roboitalia.com

BLOG FORUM GMAIL

Roboitalia.com - Il primo portale in Italia sulla robotica amatoriale > Forum di robotica avanzata > Elettronica Generale
Adc pic

Username Username Ricordami?
Password Entra

Registrazione Faq Community Calendario I messaggi di oggi Cerca

Strumenti discussione Modalità visualizzazioe

13-04-2009, 20.09.48 #1

Lin_nUx
Robottaro senior

Data registrazione: 24-08-2006
Messaggi: 87
Potenza reputazione: 16

ADC pic : Vref e tensioni negative

Salve a tutti!
Vorrei delle informazioni sul convtitore AD del PIC. In particolare desidero sapere cosa avviene se si applica un potenziale negativo (rispetto alla massa) al pin ANx. Il valore numerico corrispondente è negativo? Inoltre..se non si imposta il Vref dall'apposito pin, il Vref risultà essere quello di alimentazione del PIC?
Grazie in anticipo per le risposte!

13-04-2009, 21.43.40 #2

calo
Moderator

Data registrazione: 14-06-2006
Messaggi: 662
Potenza reputazione: 29

Citazione:
Orginalmente inviato da **Lin_nUx**
cosa avviene se si applica un potenziale negativo (rispetto alla massa) al pin ANx. Il valore numerico corrispondente è negativo?

NO [Non credo restituirà numeri negativi, ma solamente i 10bit (o più) di risoluzione dell'adc partiranno dal valore negativo di tensione.]

Citazione:
Orginalmente inviato da **Lin_nUx**
Inoltre..se non si imposta il Vref dall'apposito pin, il Vref risultà essere quello di alimentazione del PIC?

Questo lo puoi vedere sul datasheet, è infatti una delle impostazioni che si fa sul registro ADCON1 (sui 16F). Sul datasheet trovi anche lo schema interno dell'ADC, dove si nota che se non vengono dati Vref+ e Vref- dall'esterno, verrano usate Vdd e Vss. Quando dico che non vengono date le Vref dall'esterno, non vuol dire che queste non siano applicate fisicamente ai pin, ma che mediante configurazione di ADCON1 non vengono usate dall'ADC.

13-04-2009, 22.55.23 #3

astrobeed
Robottaro sostenitore

Data registrazione: 18-03-2004
Messaggi: 1,744
Potenza reputazione: 75

Non puoi applicare una tensione minore di Vss (negativo) agli ADC di un pic, il valore misurato rimane 0 ed entrano in funzione i diodi di clamping a protezione degli ingressi.
Se devi misurare una tensione AC, o una tensione negativa, devi prima applicare un offset DC che la porta nel range ammesso Vdd-Vss.
Quanto sopra solo se devi fare misure su tutto il range AC, se ti basta fare misure solo sulla porzione positiva basta mettere un R di opportuno valore in serie all'ingresso in modo da limitare a pochi mA la corrente che scorre sul diodo di clamping quando la tensione diventa negativa.

14-04-2009, 08.49.24 #4

Lin_nUx
Robottaro senior

Data registrazione: 24-08-2006
Messaggi: 87
Potenza reputazione: 16

Ringranzio entrambi per le risposte esaurienti! Altre due curiosità:
1) Al pin Vref- è quindi possibile applicare una tensione minore della Vss oppure il limite è la Vss?
2) Qual è la corrente assorbita dal pin ANx per effettuare la lettura della tensione?

14-04-2009, 09.24.58 #5

astrobeed
Robottaro sostenitore

Data registrazione: 18-03-2004
Messaggi: 1,744
Potenza reputazione: 75

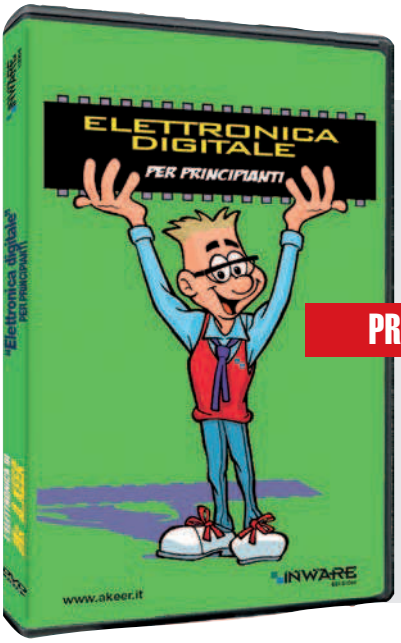
Citazione:
Orginalmente inviato da **Lin_nUx**
1) Al pin Vref- è quindi possibile applicare una tensione minore della Vss oppure il limite è la Vss?

Il limite è Vss.

Citazione:
2) Qual è la corrente assorbita dal pin ANx per effettuare la lettura della tensione?

Microchip consiglia che l'impedenza d'uscita della sorgente sia 5 kohm se vuoi ottenere la massima velocità di sampling.
Se la velocità di sampling non è una priorità, per esempio ti basta acquisire solo qualche centinaio di campioni al secondo, l'impedenza d'uscita della sorgente può essere molto maggiore dei 5 kohm.
Se ti leggi il capitolo dell'ADC sul data sheet trovi tutti i dettagli su come determinare il tempo minimo di acquisizione con tanto di formulette.

Lo shop della Community dell'elettronica



Impara l'elettronica!

Il terzo DVD della collana "Mr A.Keer" con il corso di elettronica digitale per principianti! Tra gli argomenti trattati: algebra binaria, porte logiche fondamentali, registri, contatori.

PRIMO PIANO

CODICE MAK-DIGUTALE

PREZZO: 12,00

OFFERTA

Databook

Un CDROM contenente il database con oltre 90.000 diodi e tiristori con le loro valori massimi consentiti, pin-out, dati del costruttore e dimensioni

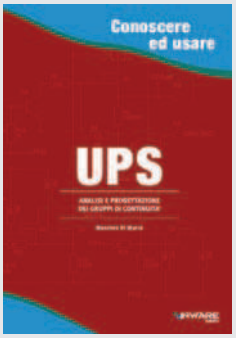
CODICE ECA-409

PREZZO: ~~€ 54.80~~ € 30.00

Nuovo libro sugli UPS

Testo dedicato alle tecniche di progettazione degli UPS alla luce delle problematiche inerenti i dusturbi della rete elettrica. Ecco l'indice degli argomenti:

NOVITA'



- classificazione degli ups.
- dispositivi di commutazione di potenza.
- tecniche di conversione cc-ac
- circuiti di controllo
- ups short break.
- inverter e ups sinusoidali
- batterie e circuiti di ricarica.
- circuiti di interfaccia e software di gestione.

Autore: Massimo Di Marco
Formato: 17x24cm
Pagine: 116

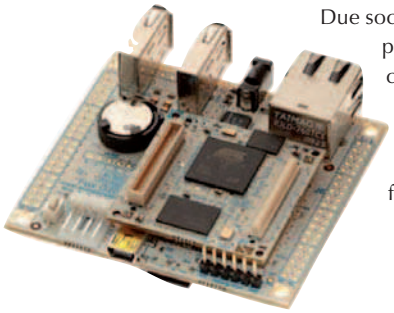
CODICE FE-41

PREZZO: 21,50

Ordina subito il libro usando la tecnologia IEmobile! I primi 10 potranno averlo a soli €19,90 comprese le spese di spedizione!

Un computer a scheda singola

FOX Board G20 è un computer a scheda singola con Linux Embedded che fa uso del core Netus G20 (Atmel AT91SAM9G20 CPU - ARM9, 400MHz, 64MB SDRAM, 8MB FLASH). FOX G20 include molte periferiche on-board quali: lettore di card uSD (card uSD non inclusa), 1 Ethernet, 2 USB 2.0, RTC, 2 porte seriali e 80 I/O GP analogico/digitale. Migliaia di applicazioni possono essere installate e compilate direttamente sulla FOX Board G20, grazie all'SDK integrato nel sistema (editor, compiler, link, library, make, etc.) in ogni linguaggio di programmazione (C, C++, Python, Perl, Java, PHP, etc.).



Due sockets 40-pin sono disponibili per avere accesso a molti I/O GP analogico/digitale, porte seriali (UART, I2C, e SPI) e interfacce PWM.

CODICE FOX BOARD G20

PREZZO: € 166.80

NOVITA'

Modulo Camera Seriale JPEG

uCAM-TTL (microCAM) è un modulo camera seriale altamente integrato che può essere collegato a qualsiasi sistema host richiedente una videocamera o una camera JPEG compressa per applicazioni visive integrate. Il modulo usa un sensore colore CMOS VGA OmniVision insieme a un chip per compressione JPEG che fornisce un sistema a basso costo e a bassa alimentazione. Il modulo ha un'interfaccia seriale TTL on-board che è usufruibile per una connessione diretta a qualsiasi host controller o a un sistema PC dotato di porta COM. I comandi utente sono spediti utilizzando un semplice protocollo seriale che può istruire la camera a mandare immagini raw a singolo frame e bassa risoluzione (160x120 o 80x60) per un veloce controllo visivo o JPEGs ad alta risoluzione (640x480 o 320x240) per storage o visualizzazione.

CODICE UCAM-TTL

PREZZO: € 54.00

Modulo display oLED

µOLED-96-PROP è un modulo display embedded estremamente compatto ed economicamente produttivo pronto all'immediato utilizzo, capace di fornire funzionalità 'stand-alone' al proprio lavoro. Il modulo rappresenta una combinazione unica con un potente micro controller, chip Propeller da Parallax Inc, display OLED 0.96" da 65K colori vivi a 96x64 pixel e slot per memory card tipo micro-SD per una capacità di storage fino a 2GB; il tutto nelle incredibili dimensioni di 27mm X 23mm.

OFFERTA

CODICE µOLED-96-PROP

PREZZO: ~~€ 76.80~~ € 58.80

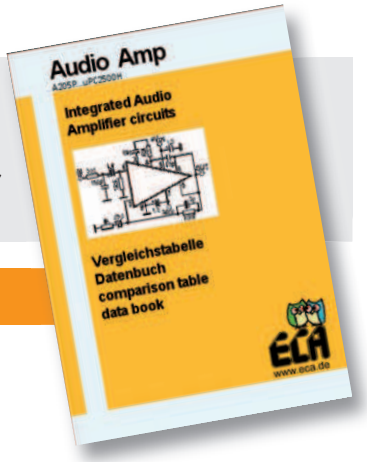
Databook

Non c'è bisogno di tante parole per presentare i data-book della ECA, ogni laboratorio che si rispetti ne possiede uno. *Audio amplifier* è un nuovo data book sui circuiti degli amplificatori audio. Oltre 3.500 circuiti diversi da A205P a µPC2500H, schemi delle connessioni, applicazioni, costruttori, equivalenze, packages, indice funzionale, tavola delle caratteristiche con tutti i parametri comuni in una tabella

CODICE AUDIOAMP

PREZZO: ~~€ 23.80~~ € 15.60

OFFERTA



Da non perdere!

Quattro libri della collana "Conoscere ed usare" ad un prezzo molto conveniente! Il bundle comprende i seguenti volumi della collana "Conoscere ed usare":
Display LCD, Programamzione Basic per PIC, Linguaggio ANSI C, UPS.

BUNDLE

CODICE: BUNDLE N

PREZZO: €69.99

Lo shop della Community dell'elettronica

BEST SELLER

Pillole di... Microcontrollori PIC

Un nuovissimo volume di ben 216 pagine dedicato alla progettazione e alla programmazione in C dei microcontrollori PIC. Il libro tratta vari modelli di PIC dal 16F877A al 18F4550.
Ecco l'indice degli argomenti:
- I microcontrollori ed i sistemi embedded
- Gli strumenti di sviluppo
- Hello word!
- Gli input digitali
- I Timer e gli interrupt
- Gli ingressi analogici
- Capture Compare PWM
- Il PIC scrive: LCD alfanumerici
- Il PIC disegna: LCD grafici
- La periferica USART
- La periferica MSSP: I2C e SPI
- USB: Universal Serial Bus
- I PIC in rete: connettività con Ethernet
- EEPROM interna, memoria FLASH e altre librerie

I numerosi esempi pratici di programmazione scritti in mikroC ti consentiranno di gestire tutte le periferiche del PIC in modo facile e veloce.

Autore: Paolo Rognoni
Formato: 17x24cm
Pagine: 216

CODICE: FE-40 PREZZO: €27.50

CUBLOC CB405

Modulo core CUBLOC CB405RT BASIC con Ladder Logic Industrial Controller. Ideale per applicazioni richiedenti largo numero di porte I/O, back-up della memoria dati durante un power-down, larga memoria dati (esempio modalità Master Modbus RTU). Ideale sia per piccole produzioni sia per produzione di massa (da 1 a 1000+), integrabile in PCB personalizzabile.

CODICE CB405RT
PREZZO: € 78.00

Kit di sviluppo RFID

Il kit di sviluppo RFID di CCS include un semplice transponder read-only e read/write per dimostrare multiple possibilità di comunicazione senza contatti diretti. Un manuale con esempi in codice sorgente spiegano come utilizzare i drivers, abilitando l'utente a sviluppare in modo veloce le proprie applicazioni RFID. La scheda prototipo RFID ha un antenna RFID a corto raggio connessa a una ricetrasmittente RFID IC. (Nota: l'antenna a corto raggio è una traccia PCB, non è un antenna collegata esternamente) La scheda prototipo delle dimensioni di 3.625" x 2" si connette a componenti esterni usando un bus RS485 four-wire.

CODICE ECL-RFID
PREZZO: € 594.00

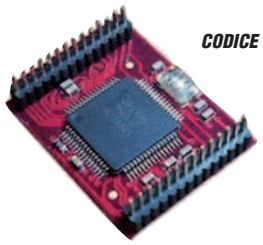
Kit di sviluppo USB Master

La scheda prototipo USB Master permette allo sviluppatore di aggiungere la funzionalità USB master a un sistema embedded senza la necessità di possedere conoscenze dettagliate del protocollo USB. La scheda prototipo ospita un chip Vinculum™ VNC1L che permette a un microcontroller di interfacciarsi con i dispositivi USB agendo da master bus.

CODICE ECL-USB
PREZZO: € 714.00

Driver VGA

Il µVGA-PICASO-MD1 è un dispositivo embedded compatto ed economico che permette la gestione grafica di ogni progetto con microcontrollore. I comandi seriali sono semplicissimi e consentono la gestione totale del display QVGA, VGA e SVGA, e' possibile controllare il colore dello sfondo, oppure creare testi di qualsiasi dimensione, colore e forma, e' anche possibile disegnare forme geometriche che includono caratteri bitmap oppure immagini precaricati dall'utente tramite il programma per PC scaricabile gratuitamente. Il tutto in 262.000 colori, senza appesantire il microcontrollore da tutte le funzioni grafiche che richiedono molte risorse.



CODICE UVGA-PICASO-MD1T
PREZZO: € 64.80

www.ieshop.it

I 10 articoli più richiesti

1- Libro "Pillole di Microcontrollori PIC"

2 - Libro "Dispositivi per l'ascolto segreto"

3 - Scheda easyPIC6

4 - Ebook "Amplificatori operazionali"

5 - Bundle "4 ebooks"

6 - Ebook "alimentatori switching"

7 - MikroBASIC-PRO

8 - MikroC-dsPIC

9 - Scheda potenziometro digitale per easyPIC

10 - Poscope, oscilloscopio USB

Ordinare direttamente tramite internet conviene! Potrai infatti aggiudicarti dei buoni sconto (coupon) con le modalità:

COUPON ABBONAMENTO ONLINE FE E FW
Sottoscrivendo un nuovo abbonamento o rinnovando il proprio online, direttamente tramite IE SHOP, si ha diritto ad un coupon del 15%! **Qualora ci si abbona (sempre online) a Fare Elettronica e Firmware insieme il coupon è del 20%!**

COUPON QUIZ "ELETTROQUIZ"
Rispondendo ai Quiz presente ogni mese su Fare Elettronica si avrà diritto ad un coupon del 10%. Tutte le risposte pervenute saranno accuratamente valutate dagli autori dei quesiti e, in caso di risposta esatta (oltre al Coupon) potrai vincere fantastici premi!

COUPON "ACQUISTI PREMIATI"
I tuoi acquisti su www.ieshop.it vengono premiati con un coupon del 10% se il tuo ordine supera i 100 EUR (IVA e spese di trasporto escluse). Ma se il tuo ordine supera i 200 EUR, lo sconto sale al 20%!

IMPORTANTE!
- Il coupon può essere speso **esclusivamente** ordinando sul sito www.ieshop.it
- Il coupon non potrà essere utilizzato in caso di nuova sottoscrizione o rinnovo dell'abbonamento, è invece utilizzabile per tutti gli altri prodotti presenti su www.ieshop.it
- Ogni coupon ha una scadenza, non dimenticarlo!
- Lo slogan "più acquisti più risparmi" pecca senz'altro di originalità, ma in questo caso è molto azzeccato: i coupon sono infatti utilizzabili solo una volta, usalo bene!
- I coupon non sono cumulabili, e vanno utilizzati singolarmente, secondo il criterio cronologico di assegnazione.

CONTI CORRENTI POSTALI - Ricevuta di Versamento — BancoPosta

sul C/C n. 70107552 di Euro ,

importo in lettere

INTESTATO A:
INWARE EDIZIONI SRL

ESEGUITO DA:

VIA - PIAZZA

CAP

LOCALITÀ

BOLLO DELL'UFF. POSTALE

AVVERTENZE
Il Bollettino deve essere compilato in ogni sua parte (con inchiostro nero o blu) e non deve recare abrasioni, correzioni o cancellature.
La causale è obbligatoria per i versamenti a favore delle Pubbliche Amministrazioni.
La mancanza o l'inesattezza anche di una sola delle indicazioni richieste in maniera non chiara e/o precisa può comportare il mancato recapito del bollettino.

CAUSALE

CONTI CORRENTI POSTALI - Ricevuta di Accredito — BancoPosta

sul C/C n. 70107552 di Euro ,

TD 451

importo in lettere

INTESTATO A:
INWARE EDIZIONI SRL

CAUSALE

ESEGUITO DA:

VIA - PIAZZA

CAP

LOCALITÀ

BOLLO DELL'UFF. POSTALE
codice bancoposta

IMPORTANTE: NON SCRIVERE NELLA ZONA SOTTOSTANTE
importo in euro — numero conto — td

AUT. DB/SISB/E 20017 DEL 11.04.2006

70107552 < 451>

IMPORTANTE: NON SCRIVERE NELLA ZONA SOPRASTANTE!

AVVERTENZE

Il Bollettino deve essere compilato in ogni sua parte (con inchiostro nero o blu) e non deve recare abrasioni, correzioni o cancellature.

La causale è obbligatoria per i versamenti a favore delle PubblicheAmministrazioni.

Le informazioni richieste vanno riportate in modo identico in ciascuna delle parti di cui si compone il bollettino.

Indagine sui lettori

Aiutaci a conoscerti meglio!
Con il tuo aiuto riusciremo ad offrirti una rivista sempre più in linea con le tue aspettative!

Nome .....

Cognome .....

Via ..... n° .....

Cap ..... Città ..... Prov .....

Email .....

IL TUO SETTORE DI COMPETENZA:

- ❑ **B05** Direzione Tecnica
- ❑ **B06** Progettazione
- ❑ **B07** Studente
- ❑ **B08** Direzione Acquisti
- ❑ **B09** Insegnante
- ❑ **B10** Altro . . . . . .

.

PRODOTTO PRINCIPALE O SERVIZIO OFFERTO DALL'AZIENDA DOVE LAVORI:

- ❑ **C11** Apparecchiature elettriche, elettroniche, ICT
- ❑ **C12** Elettrodomestici
- ❑ **C13** Consulenza
- ❑ **C14** Apparecchiature scientifiche, misura e controllo
- ❑ **C15** Automotive
- ❑ **C16** Vending
- ❑ **C17** Altro . . . . . .

NUMERO DI DIPENDENTI DELLA TUA AZIENDA:

- ❑ **D18** fino a 10
- ❑ **D19** da 10 a 50
- ❑ **D20** da 50 a 100
- ❑ **D21** da 100 a 500
- ❑ **D22** oltre 500

Solo se sei abbonato, indica il tuo codice abbonato: . . . . . .
e barra la casella di interesse:

TIPO DI ABBONAMENTO:

- ❑ **A01** Personale uso professionale
- ❑ **A02** Aziendale
- ❑ **A03** Scuola o Università
- ❑ **A04** Personale uso hobbistico

Compila il cedolino e invialo in busta chiusa o via fax allo 02 66508225 e riceverai GRATIS a tua scelta UN CD-ROM DEL VALORE DI 10 EURO

FOTOGRAFIA DIGITALE: un corso completo che guida l'utente nel mondo della fotografia utilizzando le moderne fotocamere digitali.

BABYLON: il traduttore istantaneo italiano/inglese inglese/italiano più conosciuto nel mondo.

iPOD CONVERTER: il software per convertire i filmati e i file audio in un formato idoneo all'iPOD.

Grazie per la preziosa collaborazione!

PRODOTTI

Se vuoi ordinare "offline" ovvero senza usare internet, usa il seguente modulo d'ordine **PRODOTTI**

COD. PRODOTTO	DESCRIZIONE	PREZZO UNITARIO	Q.tà	Totale
			SUB TOTALE	
			SPESE DI SPEDIZIONE	
			TOTALE	

Nota. Se le righe non sono sufficienti, utilizzare più copie del modulo.
Per maggiori dettagli sulle spese di spedizione e i metodi di pagamento vedi retro cartolina.

FE 2947295

METODI DI PAGAMENTO

✸ **BONIFICO BANCARIO** ✸ **BOLLETTINO POSTALE** ✸ **CONTRASSEGNO**

☐ MasterCard ☐ VISA ☐ AMERICAN EXPRESS ☐ postepay

TITOLARE. . . . . .
NUMERO . . . . . .
DATA DI SCADENZA. . . . . CODICE DI SICUREZZA . . . . . .

INVIARE IL MODULO D'ORDINE A: Inware Edizioni srl, via Cadorna 27/31 - 20032 Cormano (MI) - oppure via FAX allo 02 66508225
(Inviando questo modulo si accettano i termini e le condizioni riportate a tergo)

Nome . . . . . Cognome . . . . . .
Indirizzo . . . . . n° . . . .
Cap . . . . . Città . . . . . Prov . . . . .
Tel . . . . . Fax . . . . . Email . . . . . .
Ragione Sociale . . . . . .
Codice fiscale . . . . . Partita IVA . . . . . .

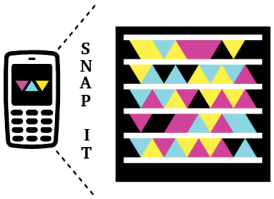
DATA . . . . . FIRMA . . . . . ✸ Fattura

ABBONAMENTO

Sì, mi abbono a Fare Elettronica

FE 2947295

- ✸ Abbonamento TRIAL a 3 numeri di FARE ELETTRONICA a soli € 15,00 anzichè € 18,00
- ✸ Abbonamento Premium a 11 numeri di FARE ELETTRONICA a soli € 49,50 anziché € 66,00
- ✸ Abbonamento PRO a 11 numeri di FARE ELETTRONICA include il CD dell'annata 2009 a soli € 59,50 anziché € 96,00



Usa IEmobile! Abbonati al PRO al prezzo del PREMIUM

METODI DI PAGAMENTO (per maggiori dettagli vedi retro cartolina)

✸ **BONIFICO BANCARIO** ✸ **BOLLETTINO POSTALE** ✸ **ALLEGO ASSEGNO** (Intestato a Inware Edizioni)

☐ MasterCard ☐ VISA ☐ AMERICAN EXPRESS ☐ postepay

TITOLARE. . . . . .
NUMERO . . . . . .
DATA DI SCADENZA. . . . . CODICE DI SICUREZZA . . . . . .

INVIARE IL MODULO D'ORDINE A: Inware Edizioni srl, via Cadorna 27/31 - 20032 Cormano (MI) - oppure via FAX allo 02 66508225
(Inviando questo modulo si accettano i termini e le condizioni riportate a tergo)

Nome . . . . . Cognome . . . . . .
Indirizzo . . . . . n° . . . .
Cap . . . . . Città . . . . . Prov . . . . .
Tel . . . . . Fax . . . . . Email . . . . . .
Ragione Sociale . . . . . .
Codice fiscale . . . . . Partita IVA . . . . . .

DATA . . . . . FIRMA . . . . . ✸ Fattura

ISTRUZIONI PER L'ORDINE E SPESE DI SPEDIZIONE

Il modulo d'ordine dovrà essere compilato in tutte le sue parti ed inviato via posta o via fax ai recapiti indicati sul modulo d'ordine stesso. Gli ordini potranno essere fatti anche direttamente online dal sito www.ieshop.it. In questo caso non sarà necessario inviare il modulo d'ordine. Le spese di spedizione ammontano a euro 8,50 a cui vanno aggiunti euro 3,50 se si sceglie di pagare in contrassegno.

METODI DI PAGAMENTO

Si accettano pagamenti ino in contrassegno, carta di credit, bollettino postale o bonifico all'ordine. Per il contrassegno verrà applicata una spesa aggiuntiva di euro 3,50 per le spese di contrassegno. Forme diverse di pagamento devono essere previamente concordate.

Bonifico bancario

Appoggiato su Poste Italiane IBAN: IT 68 I 07601 01600 000070107552 intestato a Inware Edizioni srl.

Bollettino postale

Versamento sul conto corrente postale n. 70107552 intestato a Inware Edizioni srl. Inviare la ricevuta (o copia) del versamento.

Contrassegno

La merce verrà pagata direttamente al corriere alla consegna della merce. Il pagamento in contrassegno comporta l'addebito di euro 3,50per spese di contrassegno.

Carta di Credito

Il pagamento con carta di credito può essere effettuato specificando i dati della carta via posta, email, web, fax o telefono. Per gli ordini effettuati sul sito il pagamento mediante carta di credito verrà effettuato attraverso una connessione sicura SSL per garantire la massima sicurezza.

TERMINI E CONDIZIONI

Tutti i prodotti sono coperti da garanzia. La garanzia copre tutti i difetti di fabbricazione entro un anno dal ricevimento della merce. Tutti i prodotti non funzionanti per uso improprio o incuria non saranno ritenuti in garanzia e saranno addebitati gli eventuali costi di riparazione. Tutti i prodotti verranno riparati e/o sostituiti direttamente dal produttore. Non sono coperti da garanzia i componenti elettronici (microprocessori, memorie, ecc.) La garanzia dei prodotti si intende F.co ns. sede, le eventuali spese di trasporto sono a carico del cliente salvo accordi diversi. Per dar corso alla riparazione/sostituzione in garanzia è necessario seguire l'apposita procedura di RMA.

PRIVACY

Ai sensi del Decr. Lgs. 196/2003 la informiamo che i dati trasmessi verranno impiegati coi principali scopi di indagini di mercato e nelle modalità previste dallo stesso, prevalentemente con mezzi informatici. Il conferimento, di norma facoltativo, è obbligatorio per permettere il rapporto commerciale. È in ogni caso fatto diritto dell'interessato esercitare i propri diritti, nei modi previsti dal "Titolo II art. 7" della legge sopra citata, scrivendo a Inware Edizioni srl, Via Cadorna 27 – 20032 Cormano o tramite email all'indirizzo info@inwaredizioni.it

---

ABBONARSI ALLE RIVISTE INWARE EDIZIONI CONVIENE!

I vantaggi per gli abbonati sono, oltre al prezzo bloccato per un anno, la ricezione del numero direttamente a casa con la garanzia di ricevere tutti i numeri.
Inoltre un vistoso risparmio che, nel caso dell'abbonamento plus, ammonta a ben 36,50 euro.
L'abbonamento ha una durata di 12 mesi e comporta l'invio di 11 numeri di Fare Elettronica. Eventuali variazioni di indirizzo andranno comunicate tempestivamente alla Redazione che provvederà a registrare il cambiamento senza alcuna spesa aggiuntiva. L'abbonamento decorrerà dal primo numero raggiungibile alla data di avvenuto pagamento. Non sono previsti rimborsi in caso di disdetta dell'abbonamento.

METODI DI PAGAMENTO

Bonifico bancario

Appoggiato su Poste Italiane IBAN: IT 68 I 07601 01600 000070107552 intestato a Inware Edizioni srl.

Bollettino postale

Versamento sul conto corrente postale n. 70107552 intestato a Inware Edizioni srl. Inviare la ricevuta (o copia) del versamento indicando nella causale:
"Abbonamento Fare Elettronica"

Carta di Credito

Il pagamento con carta di credito può essere effettuato specificando i dati della carta via posta, email, web, fax o telefono. Per gli ordini effettuati sul sito il pagamento mediante carta di credito verrà effettuato attraverso una connessione sicura SSL per garantire la massima sicurezza.

Assegno bancario

E' possibile spedire un assegno bancario insieme a questo coupon.

in edicola

L'ELETTRONICA DI
MR A.KEER

in abbinamento a

FARE ELETTRONICA & FIRMWARE

da non perdere €7,90

3 DVD a soli

nella confezione troverai 2 buoni sconto del valore di € 60 per l'acquisto di una EasyPic6 e un POSCOPE Bundle!

CODICE MIP 2779022

LA SOLUZIONE GSM/GPRS MODULARE PER OGNI NECESSITA'

PRESTAZIONI SOFTWARE UNICHE!

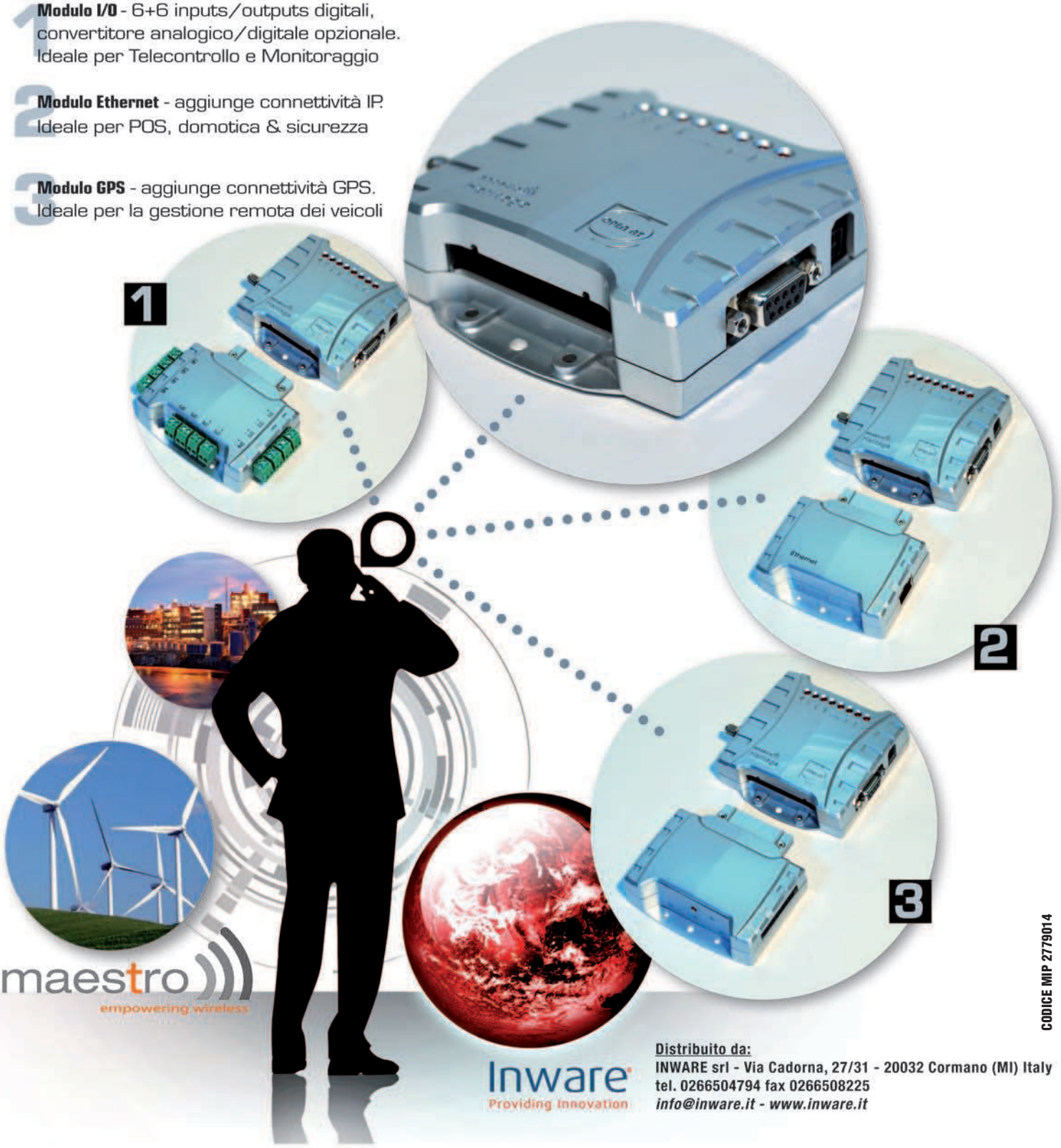
• Connessione TCP/UDP automatica • Comandi AT remoti attraverso SMS • Comandi AT con trigger sugli I/O • Controllo dello stato del modem Modem e monitoraggio • Funzionalità di aggiornamento del programma da remoto • Script per la programmazione veloce (uso del timer, allarmi, input port, contatore) • **Gestione del PING** - funzione di ping automatica per controllare e mantenere attiva la connettività GPRS. Se il ping fallisce, il modem si ricollega automaticamente o si resetta automaticamente • **Dynamic DNS** - è possibile utilizzare indirizzi IP dinamici e raggiungere il dispositivo attraverso un domain name fisso (URL). Servizio utilizzabile attraverso DDNS service providers (dyndns.com, no-ip.com...) • **TCP terminal** - è possibile accedere a Maestro Heritage via Telnet ed inviare comandi AT da remoto.

ALL'UNITÀ PRINCIPALE DI MAESTRO HERITAGE È POSSIBILE AGGIUNGERE ALTRI MODULI PER L'AGGIUNTA DI FUNZIONALITÀ:

1 **Modulo I/O** - 6+6 inputs/outputs digitali, convertitore analogico/digitale opzionale. Ideale per Telecontrollo e Monitoraggio

2 **Modulo Ethernet** - aggiunge connettività IP. Ideale per POS, domotica & sicurezza

3 **Modulo GPS** - aggiunge connettività GPS. Ideale per la gestione remota dei veicoli



CODICE MIP 2779014

Distribuito da:
INWARE srl - Via Cadorna, 27/31 - 20032 Cormano (MI) Italy
tel. 0266504794 fax 0266508225
info@inware.it - www.inware.it